

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Disseny i desenvolupament d'un joc educatiu 3D amb Unity

MEMÒRIA

Autors: Jose Antonio Alcaide Jiménez
Marc Sebastián Padrós

Director: Marc Vigo Anglada

Convocatòria: Juliol 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

L'objectiu del treball és, per una banda, desenvolupar un joc en tres dimensions, i per altra banda, que sigui un joc educatiu que motivi als usuaris a aprendre una assignatura universitària. Els reptes que els jugadors es trobin es resoldran mitjançant un procés d'hipòtesi, prova i reflexió sobre el món simulat dins del joc.

Per dur a terme aquest projecte, primerament se'n va establir l'organització i la planificació per a tota la durada i es van emprar les eines *web* de *Trello* i *Google Drive*. Seguidament es va realitzar el disseny de les mecàniques de joc i es va definir l'estil gràfic del joc.

El desenvolupament del projecte es va realitzar fent servir el repositori de *Unity* per treballar de manera conjunta en el codi. A més, es van incorporar al projecte arxius *XML* per desar dades del joc i de l'usuari i es va realitzar la interfície d'usuari amb *software* d'edició gràfica.

El resultat és un videojoc tridimensional en tercera persona, amb llibertat de moviment i d'exploració en un món fictici. El jugador interactuarà amb l'entorn mentre resol tests interactius. Aquests tests tenen relació amb l'assignatura d'estudi i són necessaris per poder avançar i progressar.

En conclusió, l'estudi de la validació ha mostrat l'acolliment del joc en un entorn real de jugadors. Els usuaris asseguren haver adquirit coneixements de la matèria mentre jugaven. A més a més, la majoria descriuen l'experiència com a divertida i útil, la qual cosa corrobora les bases del projecte. Per tant, pensem que la gamificació a l'ensenyament constitueix una eina amb un gran potencial. De cara al futur, pot ser interessant la utilització de jocs educatius en l'àmbit universitari.

Sumari

RESUM	3
SUMARI	5
1. GLOSSARI	9
2. PREFACI	11
2.1. Motivació	12
3. INTRODUCCIÓ	13
3.1. Objectius del projecte	13
3.2. Abast del projecte	13
4. ESTAT DE L'ART	15
4.1. Camp empresarial	15
4.2. Camp de la salut	15
4.3. Camp de l'ensenyament	17
4.4. Públic Objectiu	18
5. DISSENY CONCEPTUAL	20
5.1. Descripció General	20
5.2. Mecàniques de joc	20
5.3. Objectius d'aprenentatge	23
5.4. Disseny de nivells	23
5.4.1. Cova	24
5.4.2. Bosc	25
5.4.3. Poble	25
5.4.4. Cementiri	25
5.4.5. Port	26
5.4.6. Castell	26
5.5. Disseny de tests	26
5.5.1. Portes Bàsiques	26
5.5.2. Circuits Lògics	27
5.5.3. Taules de la veritat	27
5.6. Interfície d'usuari	28
5.6.1. Finestra d'inici	28
5.6.2. Menú de pausa	29
5.6.3. Heads-Up Display (<i>HUD</i>)	32
6. DECISIONS DE DISSENY	34

6.1. Motor de videojocs.....	34
6.2. Recursos artístics	35
6.3. Recursos d'organització.....	35
6.3.1. Trello.....	36
6.3.2. Google Drive:.....	37
6.3.3. Unity Collaboration:.....	37
7. INTRODUCCIÓ A UNITY	38
7.1. Assets	38
7.2. Scenes	38
7.3. GameObject.....	38
7.4. Components.....	39
7.5. Prefabs.....	40
7.6. Editor de Unity	40
7.7. Editor de Unity	41
8. IMPLEMENTACIÓ DEL DISSENY	43
8.1. Pantalla de joc	43
8.2. Personatge Principal.....	45
8.2.1. Model del personatge	45
8.2.2. Character Controller.....	46
8.2.3. Màquina d'estats.....	47
8.2.4. Player Controller	49
8.2.5. Càmera.....	50
8.3. Interaccions amb l'entorn i missions	52
8.3.1. Missions.....	52
8.3.2. Objectes.....	53
8.3.3. Tòtems.....	55
8.3.4. NPC	58
8.4. Tests interactius.....	60
8.4.1. Finestres de test	61
8.4.2. Funcionament dels tests	62
8.5. Nivells del joc	63
8.5.1. Cova	63
8.5.2. Bosc.....	64
8.5.3. Poble.....	65
8.5.4. Cementiri	66
8.5.5. Port	66
8.5.6. Castell.....	67

8.6. Menú d'inici.....	67
8.7. Menú de Pausa	68
8.7.1. Finestra d'inici	69
8.7.2. Arxiu.....	70
8.7.3. Missions.....	73
8.7.4. Mapa.....	75
8.7.5. Inventari	76
8.7.6. Menú d'opcions.....	77
8.8. Música i Efectes de So	79
8.9. Controls del Joc.....	79
8.10. Emmagatzematge de dades	80
9. VALIDACIÓ	82
9.1. Intenció de la validació	82
9.2. Qüestionari	82
9.3. Resultats.....	85
10. REPTES I DIFICULTATS DEL PROJECTE	90
11. FUTURES MILLORES	92
12. PLANIFICACIÓ I PRESSUPOST	93
13. IMPACTE AMB L'ENTORN	95
CONCLUSIONS	96
AGRAÏMENTS	97
BIBLIOGRAFIA	98
ANNEX	103

1. Glossari

Script: Guió o conjunt d'instruccions informàtiques.

Frame: Unitat mínima en la qual es pot descompondre una seqüència de video.

Textura: Imatge utilitzada per cobrir una superfície d'un objecte virtual.

Màquina d'estats: Sèrie d'estats i transicions d'estats subjectes a diferents condicions.

Mecàniques de joc: Conjunt de regles i característiques que formen un joc, com per exemple el moviment del personatge i les seves possibles interaccions amb l'entorn.

Motor de videojocs: Programari orientat al desenvolupament de videojocs.

Unity: Motor de videojocs multi-plataforma desenvolupat per *Unity Technologies*.

Asset: Elements i fitxers introduïts a *Unity* com imatges, textures i scripts, entre d'altres.

Classe: Model o construcció que permet agrupar un conjunt de variables i mètodes.

C# (C sharp): Llenguatge de programació orientat a objectes.

GameObject: Objectes de *Unity* als quals es poden afegir *Components* per modificar-ne les característiques.

Component: Eines de *Unity* que es poden afegir als *GameObjects*.

UI o GUI (Interfície d'usuari): Fa referència als menús i botons que existeixen en un programa informàtic.

XML (eXtensible Markup Language): Format de fitxer mitjançant el qual es poden emmagatzemar dades de forma llegible.

Popup: Finestra emergent.

2. Prefaci

Al llarg dels últims anys, el terme gamificació ha guanyat una atenció important tant entre la indústria [1] com entre els acadèmics [2]. La gamificació es defineix com l'ús dels elements i de les mecàniques de jocs en contextos aliens a aquest, amb l'objectiu d'orientar el comportament de les persones i aconseguir determinades fites [3].

El terme gamificació va aparèixer per primer cop l'any 2002 [4], però no va ser fins al 2010 que es va definir en un sentit més específic amb la incorporació de recompenses i aspectes socials als jocs [5].

Una recopilació d'investigacions sobre la gamificació en mostra els efectes positius sobre les persones [6]. D'aquesta manera, la gamificació pot millorar la capacitat d'un individu per realitzar diverses tasques, com és ara millorar el compromís de l'usuari [7], la productivitat organitzativa [8] i l'exercici físic [9], entre d'altres. En concret, la gamificació té una gran importància en el camp de l'aprenentatge acadèmic [10] i pot ajudar a aprendre continguts digitals i comprendre una determinada àrea d'estudi, com per exemple, l'electrònica.

La gamificació es pot aplicar fent servir diverses eines: jocs de taula, cartes, etc. No obstant això, els videojocs constitueixen el medi més flexible per proporcionar un entorn d'aprenentatge. Aquests jocs electrònics reuneixen imatge, so i estructura literària, de la mateixa manera que ho fan les novel·les i les pel·lícules. A més a més, afegixen una característica que els és única: la interactivitat permet a l'usuari ser el protagonista de l'experiència i per tant, influir-hi.

Actualment, la indústria dels videojocs és un sector innovador que està en creixement continu i Espanya és un dels països en els quals hi ha més consum en videojoc. Segons l'Associació Espanyola de Videojocs (AEVI), aquest sector ha facturat 1163 milions d'euros l'any 2016, un 7,4% més que l'any 2015, i gairebé el doble que la indústria del cinema [11].

L'entorn d'aprenentatge proporcionat per un videojoc educatiu permet als estudiants adquirir coneixements per tal de resoldre problemes de major complexitat. Així, els reptes dins d'un joc són inicialment senzills i progressivament es fan més difícils a mesura que es desenvolupen les habilitats dels jugadors.

2.1. Motivació

Abans d'iniciar el grau, ja reconeixíem la informàtica com una ciència apassionant i amb molt potencial. El Grau en Enginyeria en Tecnologies Industrials ens va proporcionar l'entorn d'aprenentatge necessari per poder familiaritzar-nos amb la programació i les seves aplicacions.

Concretament, les assignatures *Fonaments d'informàtica* i *Informàtica* ens van permetre adquirir la base teòrica necessària per aprendre qualsevol llenguatge de programació. A més a més, vam tenir l'oportunitat d'aplicar aquests coneixements en diverses assignatures optatives com *Jocs per a Computadors*, mitjançant la realització d'un joc [12], *Halimdor's Curse* i *Projecte II*, on es va desenvolupar *Lord of Math* [13], una aplicació per a *Android* per ajudar a infants amb les matemàtiques.

De l'experiència adquirida en el projecte de *Lord of Math*, vam poder comprovar el gran potencial dels jocs com a eina educativa. Quan el joc no és només una successió unidireccional de tests, sinó que proporciona al jugador les eines i llibertat, s'obre un món de possibilitats per involucrar experiències que ajuden en algun aspecte als jugadors. Motivats pels resultats del projecte, vam voler realitzar un joc educatiu de més complexitat i en tres dimensions.

A més a més, tots els nostres coneixements fins ara en l'àmbit de la programació es limitaven a *Python*, el llenguatge que hem après a la universitat. A causa del nostre interès per continuar aprenent en l'àmbit de la informàtica i el desenvolupament no només de videojocs, sinó de *software* en general, vam pensar que seria una bona idea aprofitar el treball de fi de grau per aprendre un nou llenguatge i alhora aprofundir en diversos aspectes que ens interessin, com és ara la creació d'un joc en tres dimensions, que afegeix molta complexitat al desenvolupament.

3. Introducció

3.1. Objectius del projecte

L'objectiu del projecte és dissenyar, implementar i validar un joc seriós en 3D per ordinador orientat a l'aprenentatge d'un camp de l'enginyeria. Concretament, es pretén facilitar l'ensenyament del camp de l'electrònica a estudiants mitjançant la resolució de puzles interactius amb el món virtual.

De manera general, l'objectiu és aconseguir un joc que motivi els jugadors per aprendre. Els reptes que els usuaris es trobin es resoldran mitjançant un procés d'hipòtesi, prova i reflexió sobre el món simulat dins del joc. Els jugadors disposaran en tot moment d'un objectiu clar i tota la informació docent necessària per a assolir cada objectiu. La informació proporcionada incrementarà la seva complexitat gradualment i serà vàlida tant per a inicialitzar-se amb l'àmbit d'estudi com per a revisar els coneixements adquirits.

Seguint aquests principis, es desenvoluparà un joc on la interpretació de la informació es converteix en un objectiu intrínsec del joc. El compromís, l'experimentació i la resolució de reptes difícils seran les bases per a l'aprenentatge.

De manera específica, amb el següent projecte es crearà un joc capaç d'aplicar tots els coneixements informàtics adquirits en una branca de la tecnologia treballada durant els estudis universitaris. L'àmbit escollit és l'electrònica, específicament els circuits i portes lògiques. Aquest camp proporciona la varietat necessària per poder presentar al jugador reptes amb diversitat i complexitat creixent.

Com a objectiu secundari del projecte, destaquem l'aprenentatge del procés de desenvolupament de software, l'aprenentatge d'un nou llenguatge de programació, *C#*, i d'una eina informàtica moderna de disseny de jocs, *Unity*.

3.2. Abast del projecte

El present projecte engloba totes les fases de disseny, implementació i validació del joc seriós descrit en l'apartat anterior. El disseny del joc inclou idear la transició de finestres (en anglès, *mock-up*), el disseny de la narrativa del joc, els nivells, les missions i els tests interactius, així com la cerca i adaptació de recursos multimèdia, com per exemple models 3D, sons, música i imatges.

La implementació del projecte inclourà el desenvolupament de tots els aspectes del joc, des del propi codi, programació del moviment i col·lisions fins a les interaccions específiques del personatge amb diferents elements del joc (tant objectes com personatges no jugables). Les mecàniques de joc, la interfície d'usuari i els tests interactius també formen part de la implementació del joc.

En la implementació del joc també s'engloba el modelatge de nivells i la integració amb la música i els efectes de so. Finalment, s'incorporarà un sistema d'emmagatzematge de dades perquè el jugador pugui reprendre una partida que ja hagi començat anteriorment.

Tanmateix, a causa del temps limitat de duració del projecte, apareix la dificultat de la realització d'un joc sencer des de zero. Per aquest motiu, s'estableixen limitacions en la planificació, tot simplificant les mecàniques de joc i descartant la inclusió de cinèmiques. Sempre que ha estat possible, s'han aprofitat recursos de lliure disposició a internet (personatges, entorns visuals, sons i música, textures, etc.) per tal d'accelerar el procés de modelatge. Generalment s'han obtingut aquests recursos gratuïtament de la tenda virtual *Unity Asset Store* [14], del portal de personatges i animacions *Mixamo* [15] i de la pàgina de distribució d'elements artístics per a videojocs *Open Game Art* [16].

El resultat final serà un joc totalment funcional amb un conjunt d'entorns diferents, en els quals el jugador tindrà certa llibertat de moviment i d'exploració. Durant el transcurs del joc, el jugador haurà de respondre preguntes relacionades amb l'electrònica, que li permetran avançar a altres nivells. Aquestes preguntes augmentaran de dificultat segons es progressa en el joc, de tal forma que suposi un repte.

Finalitzada la implementació del joc, es realitzarà una prova de validació a un grup d'usuaris de perfils diferents. L'extracció i anàlisi d'aquests resultats permetrà establir futurs àmbits de millora i determinar la validesa del projecte.

L'aprenentatge comprèn un tema de l'electrònica, la lògica. Com que es requereixen molts recursos ensenyar una assignatura sencera de forma interactiva en un videojoc, hem preferit centrar-nos en una part específica, però ensenyar tot el seu contingut. Els temes que s'han triat han estat les portes i circuits lògics de diferents dificultats i les taules de la veritat.

4. Estat de l'art

Abans d'iniciar el procés de disseny del joc, s'ha realitzat un estudi de mercat dels antecedents més rellevants en gamificació. Gràcies a la gran diversitat de possibilitats que ofereixen els videojocs, existeixen molts sectors en què aquesta tecnologia s'està aplicant de manera satisfactòria. A continuació, s'exposen alguns exemples del camp empresarial, sanitari i de l'ensenyament.

4.1. Camp empresarial

En el món empresarial, la gamificació presenta diversos avantatges, com per exemple, crear una competència sana entre els equips i encoratjar la fidelització dels clients. *Gamelearn* és la companyia líder en la utilització dels videojocs per a formació d'empreses amb més de 15 anys d'experiència [17][18]. Actualment, el seu joc més complet és *Pacific* [19], un videojoc en línia sobre lideratge i gestió d'equips. En aquest joc els usuaris aprenen i posen en pràctica estratègies, tècniques i eines per desenvolupar i millorar les seves habilitats de lideratge i gestió d'equips.

Està ambientat en una aventura de supervivència en la qual l'alumne ha d'escapar, juntament amb el seu equip, d'una illa deserta on han quedat atrapats després d'un accident aeri. En el joc es valora la capacitat per gestionar i motivar l'equip mitjançant la selecció de respostes i l'assignació de recursos. Cada llicència del joc té un cost de 40 € [20]. A la Figura 4.1 es poden veure imatges del joc.

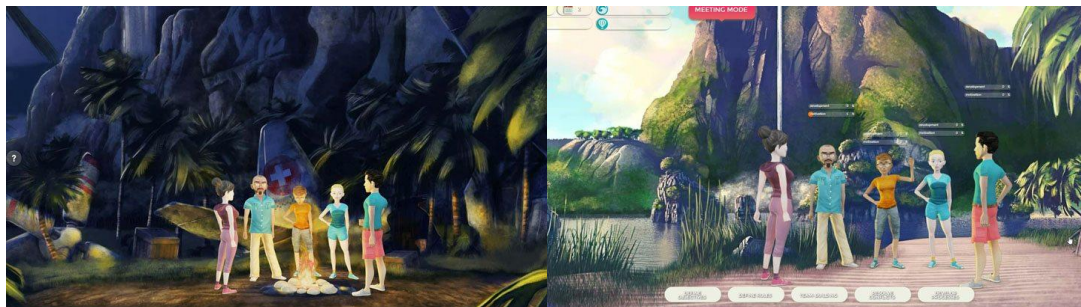


Figura 4.1 Imatges del joc educatiu *Pacific*

4.2. Camp de la salut

En el camp de la salut, la gamificació permet animar els usuaris a millorar el seu estat de salut. Mitjançant punts o premis s'encoratja els jugadors a aprendre a gestionar tota mena de malalties, des d'afeccions cròniques i trastorns mentals comuns, fins a la prevenció i control

de les infeccions.

Monster Guard és la primera aplicació per a mòbils creada per la Creu Roja Americana i està dissenyada específicament per a nens. L'objectiu és formar nens de 7 a 11 anys sobre com preparar-se per a emergències de la vida real en un joc divertit i atractiu. Patrocinada per *Disney*, aquesta aplicació gratuïta es pot descarregar en dispositius mòbils i tauletes d'*iOS* i *Android* [21]. A la Figura 4.2 es poden veure imatges de l'aplicació.

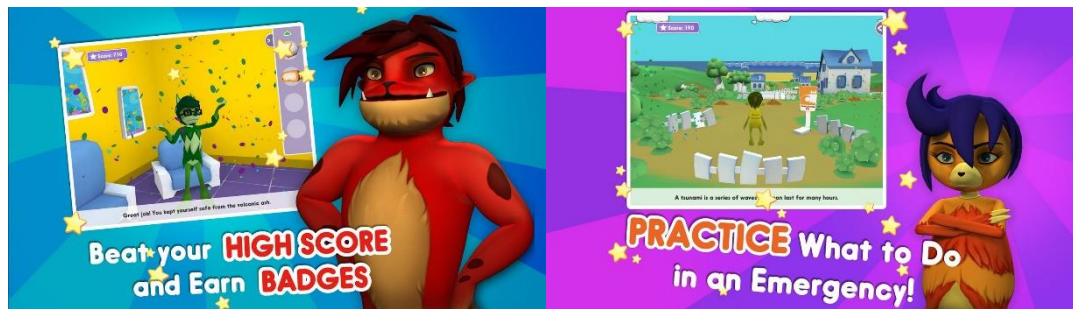


Figura 4.2 Imatges de l'aplicació *Monster Guard*

Alguns estudiants de l'ETSEIB han desenvolupat jocs educatius en el camp de la salut, malgrat que no s'han comercialitzat. Destaquen:

- **Desenvolupament d'un joc seriós per a la formació de pacients amb diabetis tipus I, Ariadna Pérez Garriga [22]:** Joc educatiu amb l'objectiu d'oferir una eina d'ajuda als pacients més joves de diabetis tipus I. Permet, a través del joc, ensenyar a comptar els hidrats de carboni dels aliments i a utilitzar aquests coneixements adquirits en el seu dia a dia. El projecte desenvolupat per *Unity* a l'ETSEIB, fa servir un entorn 3D en primera persona. A la Figura 4.3 es pot veure una imatge del joc.



Figura 4.3 Imatge del joc per la formació de la diabetis

- **Disseny i desenvolupament d'una aplicació per al tractament logopèdic, Itziar Egurbide [23]:** Joc educatiu destinat a donar suport als logopedes en la rehabilitació logopèdica d'infants que tenen trastorns lingüístics. L'aplicació implementa una de les

tasques que es duu a terme avui dia manualment a l'hospital i que consisteix en la segmentació sil·làbica. A través de l'aplicació, els nens i nenes s'entrenen en les seves habilitats lingüístiques mitjançant un sistema de reconeixement de veu. El projecte desenvolupat per *Unity* a l'ETSEIB, fa servir un entorn bidimensional. A la Figura 4.4 es pot veure una imatge del joc.



Figura 4.4 Imatge del joc per la formació logopèdica

4.3. Camp de l'ensenyament

En el camp de l'ensenyament, la gamificació s'utilitza per motivar els estudiants a aprendre. L'objectiu és maximitzar tant la diversió com el compromís a través de la captació de l'interès dels aprenents i així animar-los a continuar formant-se. Alguns exemples són *Minecraft Education Edition* [24] (plataforma per fomentar la creativitat) o *Arté Mecenas* (Joc per aprendre història) [25]

No obstant això, el joc *Variant Limits* es diferencia de la resta per ser capaç de gamificar un camp de gran complexitat: el càlcul. *Variant Limits* és un videojoc en tres dimensions per ordinador desenvolupat per l'empresa *Triseum*. L'objectiu és l'exploració d'un món 3D en el qual es fan servir exercicis de continuïtat, funcions, límits i asímptotes per tal de poder avançar.

Aquestes eines creen un entorn de treball en què el jugador pot explorar i experimentar al seu ritme. La inclusió d'una història, missions i gràfics interessants ajuden a motivar l'usuari. *Variant Limits* té un preu de 29,95\$.[26]. A la Figura 4.5 es poden veure imatges del joc.

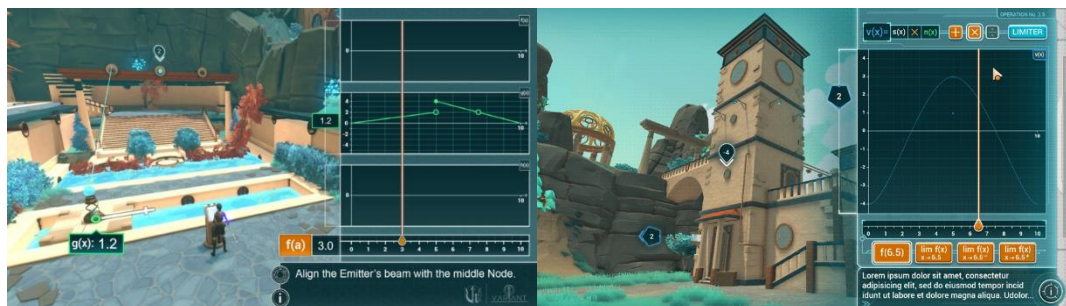


Figura 4.5 Imatges del joc Variant Limits

En una enquesta desenvolupada per *Triseum* es va obtenir que el 70% dels participants van aprendre i alhora van gaudir mentre jugaven a un joc de càlcul [27]. El joc té un gran potencial tant econòmic com educatiu en l'àmbit de la gamificació en l'ensenyament, un sector fins ara no gaire explotat. A la Figura 4.6 es pot veure un resum dels resultats de l'enquesta. *Variant Limits* és, per tant, el referent d'aquest projecte en incloure un tema educatiu complex en un videojoc divertit i motivant.



Figura 4.6 Resum dels resultats de l'enquesta.

4.4. Públic Objectiu

Atesa la gran diversitat de la gamificació en el mercat, és necessari establir el públic objectiu de l'aplicació. Principalment el projecte està orientat a dos tipus d'usuaris: universitaris que cursen electrònica i jugadors casuais de jocs d'enginy.

Per als jugadors universitaris que es troben actualment cursant electrònica, l'aplicació pot proporcionar un suport complementari a allò après a classe. D'aquesta manera els estudiants poden interactuar i experimentar lliurement amb els coneixements de l'assignatura.

El projecte també va orientat a jugadors de jocs d'enginy, ja que no és necessari interpretar els problemes com exercicis d'electrònica per gaudir del joc. Es pot plantejar el joc com la resolució d'enigmes, a la vegada que s'explora un món tridimensional. En aquest sentit, no

cal disposar de coneixements previs d'electrònica. Tanmateix, es recomana una edat mínima d'uns 16 anys i un nivell mínim d'anglès per poder entendre els continguts del joc.

La viabilitat d'aquest públic està recolzat per un informe desenvolupat per *MarketsandMarkets*. En aquesta anàlisi s'estima un creixement d'11,10 milers de milions de dòlars en el mercat de la gamificació el 2020 (el 2015 van ser 1,65 milers de milions). A més, s'espera una taxa de creixement anual compost (mesura del retorn d'una inversió) d'un 46,6% [28].

De la mateixa manera, s'espera un augment en demanda corporativa. A partir dels resultats d'una enquesta realitzada per *TalentLMS* [29], el 79% dels participants (tant estudiants corporatius com estudiants universitaris) van dir que serien més productius i estarien més motivats si el seu entorn d'aprenentatge fos més semblant a un joc. És per tant esperable un ambient molt favorable pel desenvolupament de la gamificació.

Pel que fa als requisits tècnics de l'usuari, no és necessari disposar d'experiència en jocs d'ordinador ni per gaudir del joc ni per instal·lar-lo. A més a més, el *software* no requerirà un ordinador amb especificacions molt elevades, tot i que serà necessari executar l'arxiu en el sistema operatiu *Windows*.

Finalment, resten fora de l'abast d'aquest projecte aquells usuaris que busquen un entreteniment orientat a l'acció o gràfics d'última generació. Tot i que aquest conjunt d'usuaris poden arribar a gaudir de l'experiència, no constitueixen el públic objectiu del projecte.

5. Disseny Conceptual

El primer pas per al desenvolupament d'un videojoc és el disseny conceptual, en el qual es defineixen les característiques del producte final, com per exemple els personatges, la història i les mecàniques de joc.

5.1. Descripció General

Trials of Truth és un joc educatiu en 3D del gènere *RPG* (de l'anglès: Role-Playing Game, joc de rol) disponible per al sistema operatiu *Windows*. El jugador controla un personatge en un món fictici en el qual hi ha un problema que pot acabar amb tota la civilització. L'energia de la terra està debilitant-se, i l'única manera de donar-li la força per a funcionar correctament és mitjançant l'activació dels quatre tòtems elementals que hi ha al món. Malauradament, ningú sap com activar-los i el món s'apropa al seu destí inevitablement. El joc comença amb el personatge inicial dins d'una cova, on ha caigut per accident. Dins de la cova descobreix les restes d'una civilització antiga que tenia la capacitat d'activar els tòtems. Tot seguit, el protagonista descobreix com fer-los funcionar i es disposa a trobar els tòtems elementals i així salvar la seva civilització.

5.2. Mecàniques de joc

Les mecàniques de joc són el conjunt de regles i característiques que formen un joc, com per exemple el moviment del personatge i les seves possibles interaccions amb l'entorn. A l'hora de dissenyar el videojoc, s'han d'establir les mecàniques que es volen introduir per tal de fer-lo interessant, divertit i que hi hagi un interès per part del jugador de continuar jugant.

Es tracta d'un joc en tercera persona, és a dir, la càmera del joc segueix al personatge a una certa distància, a diferència dels jocs en primera persona, en què la càmera reproduïx allò que veu el personatge. El joc disposa de moviment independent de càmera i personatge. Aquesta perspectiva presenta l'avantatge d'un camp de visió més ampli i permet entendre i interactuar més fàcilment amb l'entorn, malgrat que sacrifica el grau d'immersió que proporciona un joc en primera persona.

En relació al moviment, el personatge pot realitzar les següents accions:

- **Caminar:** En qualsevol direcció, lentament.
- **Córrer:** En qualsevol direcció, a un ritme moderat.

- **Esprintar:** En qualsevol direcció, a la màxima velocitat.
- **Saltar:** El jugador pot saltar mentre està quiet o durant el moviment, a una alçada realista per a una persona. Així i tot, pot rectificar la direcció del moviment en l'aire en qualsevol moment, ja que així es dona més control del personatge al jugador i això permet la inclusió de petits nivells de plataformes.

Al llarg de la seva aventura, el jugador es troba amb diferents elements amb els quals pot interactuar. La interacció amb els elements es realitza quan el jugador s'hi troba a prop i pot ser automàtica o prement una tecla en funció de l'element.

- **Objectes:** El jugador pot recollir objectes del terra que li poden servir per progressar en les seves missions. Els objectes són representats en el món com esferes brillants, que, en contactar amb el jugador, desapareixen i s'afegeixen a l'inventari. Alguns objectes permeten al jugador llegir una nova entrada sobre l'electrònica, mentre que d'altres són necessaris per poder continuar l'aventura.
- **Tòtem:** Element de mida lleugerament superior a la del personatge principal, que té el poder màgic de fer moure o fer aparèixer objectes, gràcies a la seva capacitat de controlar l'energia de la terra. El tòtem té dos possibles estats: Actiu i Inactiu. Segons l'estat del tòtem, els objectes es trobaran en una posició o en una altra. Per exemple, una porta pot estar oberta quan el tòtem es troba en estat actiu i tancada quan es troba en estat inactiu. Si el jugador vol passar, haurà d'interactuar amb el tòtem. Quan s'interactua amb un tòtem, apareix una finestra emergent (d'ara endavant *popup*) a la pantalla del jugador, en la qual hi ha una pregunta d'electrònica que s'ha de resoldre correctament per poder passar a la següent pregunta. Si es resolen totes les preguntes correctament, passarà a activar-se, i podria obrir-se una porta o aparèixer un objecte necessari per a alguna missió. En alguns casos, convé complir alguna condició addicional per poder fer-ho.
- **Tòtem elemental:** És un tipus específic de tòtem. N'hi ha quatre al món, un per a cada element (aigua, foc, terra i aire). L'objectiu del joc és activar-los tots.
- **NPC (De l'anglès Non-Player Character. D'ara endavant, NPC):** Personatges controlats pel computador. Quan el jugador s'apropa a un NPC, pot decidir parlar amb ell. Llavors, apareix una finestra a la part inferior de la pantalla, amb el text que representa el que li diu el NPC. Aquests acostumen a explicar alguna història referent a l'argument del joc i normalment aprofiten per demanar ajuda per mitjà d'una missió. Quan acaba la conversa, el jugador haurà començat la missió automàticament, i podrà decidir ajudar el personatge fent el que es demana a la missió. Un cop s'ha completat la missió, s'ha de tornar a parlar amb el NPC, llavors aquest li estarà molt agraït i la reputació del personatge principal haurà augmentat.

A la Figura 5.1 es pot veure el disseny inicial dels tòtems, objectes i NPCs.

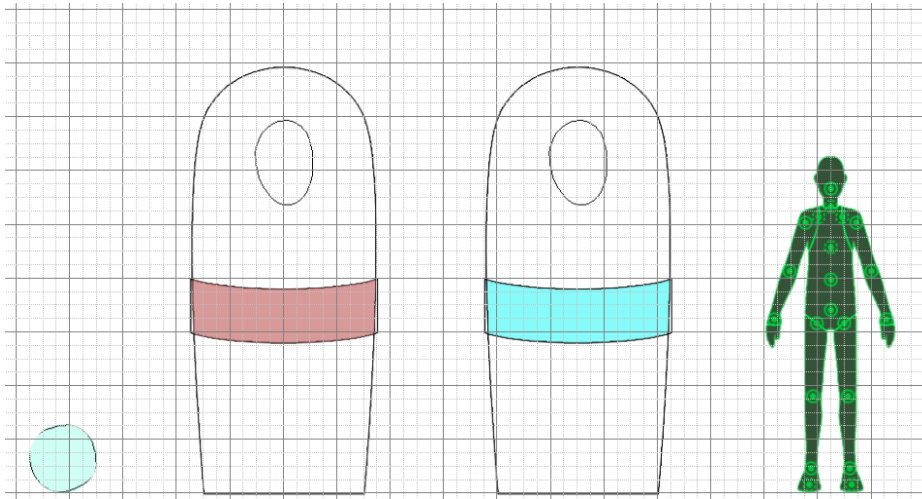


Figura 5.1 Disseny dels NPC, tòtems i objectes

Per poder avançar en l'aventura, hi ha certs punts als quals no es pot accedir, tret que es compleixin certes condicions. La condició necessària més freqüent és la puntuació de reputació. Tal com s'ha dit més amunt, aquesta s'obté sempre que es completa una missió amb èxit. Per exemple, els guardes del poble no permeten a ningú accedir al cementiri, excepte si es té una puntuació de reputació d'almenys 300 punts. Com no totes les missions tenen la mateixa dificultat ni importància, cadascuna té un premi de punts de reputació associat. Així doncs, les missions es caracteritzen en dues categories:

- **Missió Principal**
- **Missió Secundària**

Les missions són tasques que ha de realitzar el jugador, mitjançant les quals obté una recompensa de punts de reputació. Normalment, els *NPC* demanen ajuda al jugador i li assignen certes missions. Les missions principals són les tasques necessàries per a completar el joc i acostumen a obsequiar al jugador amb més punts de reputació, en general 250. Les secundàries, amb les quals s'obtenen normalment 50 punts, no cal completar-les per acabar el joc. Tot i això, el joc està dissenyat de tal manera que cal resoldre algunes missions secundàries, encara que no totes, per poder avançar.

5.3. Objectius d'aprenentatge

Ja s'ha esmentat abans que l'aprenentatge interactiu mitjançant un videojoc és un dels objectius del projecte, i que s'havia decidit que el tema fos l'electrònica. Però l'electrònica abasta molts conceptes de diferent complexitat i no és viable ensenyar una assignatura sencera en un projecte que es desenvolupa en un període de quatre mesos. Per aquest motiu, s'ha preferit triar-ne una part, per poder explicar-la amb més dedicació, i així també establir un exemple d'un joc complet amb objectius clars per a futurs projectes amb finalitats educatives. Així doncs, el joc ensenyarà als jugadors la lògica a l'electrònica i està organitzat en els següents temes:

- Portes lògiques (*Buffer, Inverter, AND, NAND, OR, NOR, XOR, XNOR*).
- Circuits amb portes lògiques amb una o més entrades i sortides.
- Taules de la veritat (en anglès: *Truth Table*).

Els *popups* que apareixen en interactuar amb els tòtems que es troben dispersats pel món, contenen tests interactius relacionats amb aquests temes. Tot i això, és necessari que el jugador tingui l'ocasió d'aprendre la teoria abans d'enfrontar-se a un test. Per aquest motiu, es disposa d'un menú anomenat *Archive* (Arxiu en anglès) que conté informació sobre totes les portes lògiques i mostra les seves respectives taules de la veritat, tot amb exemples interactius. D'aquesta manera es pot revisar el temari al qual s'hagi arribat sempre que es vulgui, ja que el jugador ha de tenir el material d'estudi disponible en tot moment. Evidentment, per tenir disponible la lectura de temari més avançat s'ha de progressar en el joc i es comença pels exercicis més senzills.

Així doncs, l'objectiu és ensenyar una part específica de l'assignatura, amb la intenció d'explicar-la amb més detall que si es fes tota una assignatura universitària sencera. Un cop s'ha triat el tema, cal decidir com s'han de dissenyar les preguntes interactives, així com el món o entorn on es troba el personatge. Aquests aspectes s'exposen al següent apartat.

5.4. Disseny de nivells

El disseny de nivells és una part fonamental del desenvolupament d'un videojoc, perquè és la part que ha de posar en pràctica totes les mecàniques de joc dissenyades i que ha d'assegurar que són atractives per al jugador. El joc es distribueix en sis zones o nivells amb diversos estils artístics i maneres de completar-los. És important planificar quin rol tenen aquests nivells i com han de ser, perquè a l'hora de desenvolupar-los no hi hagi dubtes durant la implementació del disseny i que l'experiència del jugador sigui la millor possible. Això és especialment important en el cas del nivell inicial, ja que es tracta del primer que veurà el jugador, és on s'ha de captar el seu interès i on hi ha més perill que el perdi. Aquest procediment es dificulta

si també se li ha d'ensenyar a jugar mitjançant un tutorial o similar, atès que els tutorials solen ser un procediment avorrit si el jugador ja és experimentat amb els videojocs.

A la Figura 5.2 s'exposa el disseny conceptual i artístic dels nivells que hi ha al joc, per ordre d'aparició:

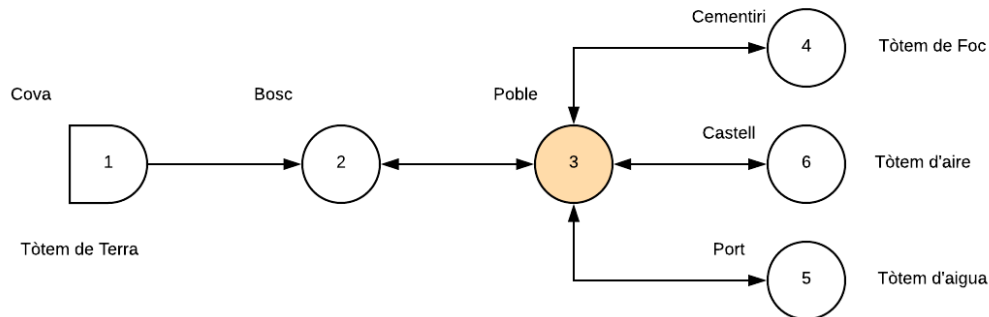


Figura 5.2 Esquema de nivells.

5.4.1. Cova

El joc comença amb el personatge principal a una cova, de la qual ha de sortir. Inicialment, el jugador no sap jugar, ni on ha d'anar; per tant és important que se li ensenyi des del primer moment com s'ha de controlar el personatge, perquè pugui caminar en direcció al seu objectiu. Per això s'ha considerat important que en aquest nivell no hi hagi llibertat en relació a on es pot accedir, de manera que no es pugui perdre. Si el nivell és lineal, el jugador no es podrà perdre, perquè hi ha només una direcció possible cap a la sortida.

El disseny artístic és, per a la primera part, una cova constituïda bàsicament per roques. Després d'aprendre els controls bàsics, el jugador descobreix una construcció d'una civilització antiga i es troba amb el primer tòtem i un passadís bloquejat per una roca, però encara no hi pot interactuar amb el tòtem per moure-la. A més, hi ha un altre passadís, a través del qual s'accedeix a una altra sala, on hi ha el primer *NPC* del joc. En parlar amb l'*NPC*, aquest posa al jugador en el context argumentatiu del joc; li explica que ha d'activar els quatre tòtems elementals que hi ha en el món i li dona un objecte mitjançant el qual tindrà la capacitat d'activar i desactivar els tòtems que es trobi pel camí. També obté nova informació a l'*Archive* sobre el temari, de manera que ja es disposi de la informació necessària per resoldre les primeres qüestions. A part, s'explica amb detall el funcionament de la interacció amb els tòtems i la resolució dels primers tests.

Un cop s'ha obtingut l'objecte necessari per a activar el primer tòtem, el jugador s'ha de dirigir

a la sala d'on venia i activar-lo mitjançant la resolució d'un test interactiu. En resoldre'l, s'aparta la roca que bloquejava el camí, de manera que ja es pot continuar el nivell. La resta del nivell consta d'una sèrie de sales amb tòtems, mitjançant les quals s'introdueixen les diferents portes lògiques a poc a poc, i s'afegeix tot el contingut après a l'*Archive*.

Al final del nivell, el jugador troba el primer dels quatre tòtems elementals que ha d'activar, en què hi ha un exercici d'una dificultat lleugerament superior a la resta. Això representa que es tracta d'un tòtem important per a la història del joc. Quan s'activa, es pot continuar a la sortida de les ruïnes de la civilització antiga i passar al següent nivell.

5.4.2. **Bosc**

Durant el segon nivell, es disposa de més llibertat per explorar i descobrir zones d'interès. Es tracta d'un gran bosc que s'ha de travessar per a arribar a un poble, que és el destí al qual s'ha d'arribar. Durant el camí hi ha un *NPC* que necessita ajuda amb l'activació d'uns tòtems per fer aparèixer uns objectes anomenats nuclis d'energia. Cal completar aquesta missió per tenir accés al poble. A part, hi ha un parell de missions secundàries que el jugador pot buscar si s'aparta lleugerament del camí. Així, és difícil perdre's si un es vol dirigir al poble, però no es renuncia al concepte d'exploració de la zona. En la mateixa línia, hi ha objectes repartits pel nivell que poden ser necessaris per a alguns altres *NPC* que el jugador es trobarà més endavant.

5.4.3. **Poble**

El poble constitueix un nivell diferent de la resta, en el sentit que no hi ha tòtems ni preguntes d'electrònica. Aquest és el punt central del joc, que unifica els accessos al bosc, al cementiri, al port i al castell a cada costat de la muralla que l'envolta. Els *NPC* habiten el poble i necessiten ajuda del jugador, que pot completar missions secundàries per tal d'obtenir punts de reputació. El poble és un punt de descans per al jugador, que aprofita per aconseguir missions, i on també se l'incentiva a retornar al bosc per completar-ne algunes i així es premia un altre cop l'exploració. La història del joc continua al cementiri, el quart nivell.

5.4.4. **Cementiri**

Per al cementiri s'ha emfatitzat en afegir unes mecàniques més sofisticades que no serien possibles en un entorn com el bosc amb molta llibertat d'exploració. Per això, es tracta d'un nivell molt lineal, en què els tòtems estan dissenyats dins d'un laberint format per les reixes metàl·liques de la tanca del cementiri. Al final del laberint hi ha un parell de tòtems addicionals que obren l'accés al segon tòtem elemental del joc. En aquest nivell també hi ha algun objecte per a una missió secundària.

5.4.5. Port

El port és similar al cementiri en el sentit que és un nivell lineal i amb una ambientació única que dificulta el disseny d'una zona gran. Es tracta d'un nivell més curt, en el qual el jugador ha de trobar el tercer tòtem elemental en un port que es troba als afores del poble. Evidentment, per a aconseguir-ho, s'han d'activar uns tòtems que tenen la capacitat d'obrir portes i fer moure objectes. Un cop s'ha aconseguit activar el tercer tòtem elemental, al jugador li quedarà activar l'últim al castell.

5.4.6. Castell

Finalment s'exposa l'últim nivell del joc, en el qual el jugador trobarà l'últim tòtem elemental. El nivell comença amb el personatge davant del castell i un pont per accedi-hi, però en posició vertical. Per col·locar-lo en posició horitzontal, cal activar un tòtem. Un cop dins, se n'han d'activar uns altres per a aconseguir 5 objectes necessaris per poder interactuar amb el tòtem elemental. A més, per poder arribar al tòtem elemental, que es troba en el sostre del castell, s'ha d'obrir una porta amb una clau que es troba dins d'una sala a la qual cal accedir mitjançant l'activació d'un altre tòtem. En aquest nivell, es realitzen tots els tipus de tests interactius d'electrònica existents en el joc a tall de repàs i també hi apareixen les preguntes més complicades.

5.5. Disseny de tests

A continuació s'expliquen els diferents tipus de tests interactius que existeixen en *Trials of Truth*, separats per les diferents categories que els constitueixen:

5.5.1. Portes Bàsiques

El tipus de test interactiu més senzill que es troba durant el joc és el de les portes bàsiques. Al *popup* que apareix en interactuar amb un tòtem, hi surt una porta bàsica, amb les seves respectives entrades i una sortida. A la Figura 5.3 es mostra un exemple d'un exercici amb una porta *Buffer*, amb una entrada a l'esquerra de la pantalla, i la sortida a la dreta. L'estat actiu (estat 1) d'una entrada o sortida estan representats de color blau cel, mentre que l'estat inactiu (estat 0) ho està de color vermell. Dins d'aquesta categoria de test s'hi diferencien dos tipus, segons allò que ha de fer el jugador:

- **1a) Modificar Entrada:** El jugador ha de fer clic amb el ratolí sobre les entrades de la porta lògica de tal manera que el valor de la sortida sigui l'estat actiu (1).

- **1b) Modificar Sortida:** Els valors de les entrades del circuit s'aleatoritzen, és a dir, qualsevol entrada pot trobar-se en l'estat inactiu (estat 0) o en l'estat actiu (estat 1). El jugador ha de clicar amb el ratolí sobre la sortida de la porta lògica, de tal manera que el valor de la sortida sigui el correcte.

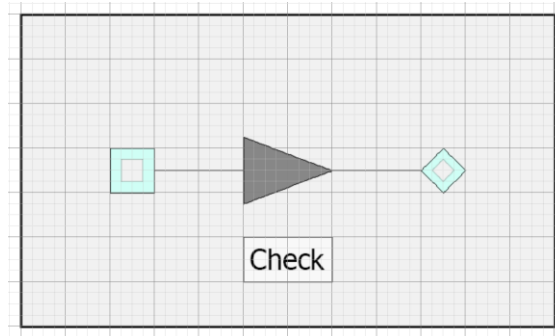


Figura 5.3 Disseny dels tests de portes bàsiques.

5.5.2. Circuits Lògics

Aquests tests són semblants a la modalitat 1b que s'acaba de mencionar, però en comptes de tractar-se d'una porta lògica, hi ha un circuit lògic amb més d'una porta lògica i diverses entrades. Es distingeixen dos tipus d'exercici:

- **2a) Sortida única:** El jugador ha de fer clic a l'única sortida que hi ha en el test i trobar el resultat correcte per a resoldre'l.
- **2b) Sortides múltiples:** En la mateixa línia, aquests tests són circuits lògics més complicats, amb múltiples entrades i sortides. El jugador ha d'encertar totes les sortides del circuit per poder resoldre el test correctament.

5.5.3. Taules de la veritat

L'última modalitat de test interactiu que hi ha en el joc és la de les taules de la veritat. Aquestes representen un sistema lògic qualsevol, aleatoritzat, que es regeix per una expressió basada en els mintermes o maxtermes del circuit. Pot haver-hi dues o tres entrades, i per tant la taula té quatre o vuit valors que ha d'assignar el jugador correctament. El *popup* mostra doncs una taula amb tres o quatre columnes, segons si hi ha dues o tres entrades, on l'última s'anomena 'Y' i representa la sortida, i les altres s'anomenen en ordre alfabètic. A l'altre costat del *popup* es mostra l'expressió per la qual es regeix el sistema a resoldre. A la Figura 5.4 es pot veure un exemple de com és un test de taules de la veritat.

A	B	Y		
0	0	0		A'B
0	1	0		
1	0	0		
1	1	0	Check	

Figura 5.4 Disseny dels tests de les taules de veritat.

5.6. Interfície d'usuari

Més enllà de la interacció al món tridimensional existeixen un seguit de finestres i pestanyes que permeten al jugador rebre i interactuar amb la informació del mateix joc. Tot el sistema de menús i finestres s'anomena interfície d'usuari o *UI* de l'anglès *User Interface* (d'ara endavant, *UI*).

5.6.1. Finestra d'inici

La primera finestra a iniciar el joc és la pantalla d'inici. El menú d'inici està format per 5 opcions, les quals es poden visualitzar a la Figura 5.5.

- **Continuar (*Continue*)**: Per continuar l'última partida que s'hagi jugat anteriorment.
- **Nova partida (*New Game*)**: Aquesta opció permet crear un nou usuari i assignar-li un nom que no hagi estat utilitzat anteriorment en l'ordinador.
- **Carregar partida (*Load Game*)**: Permet carregar una partida anteriorment desada.
- **Opcions (*Settings*)**: Obre el menú d'opcions per modificar el nivell de la música i dels efectes de so de manera independent. A més, inclou Informació sobre els desenvolupadors del joc.
- **Sortir del joc (*Exit game*)**: Opció per sortir del joc.

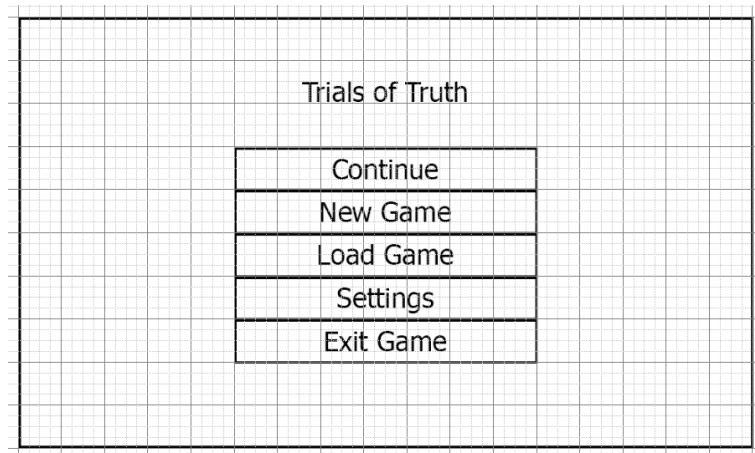


Figura 5.5 Disseny de la finestra d'inici.

La Figura 5.6 és una representació de totes les accions que el jugador pot efectuar dins d'aquest menú, en el qual es poden veure les possibles transicions en tot el joc:

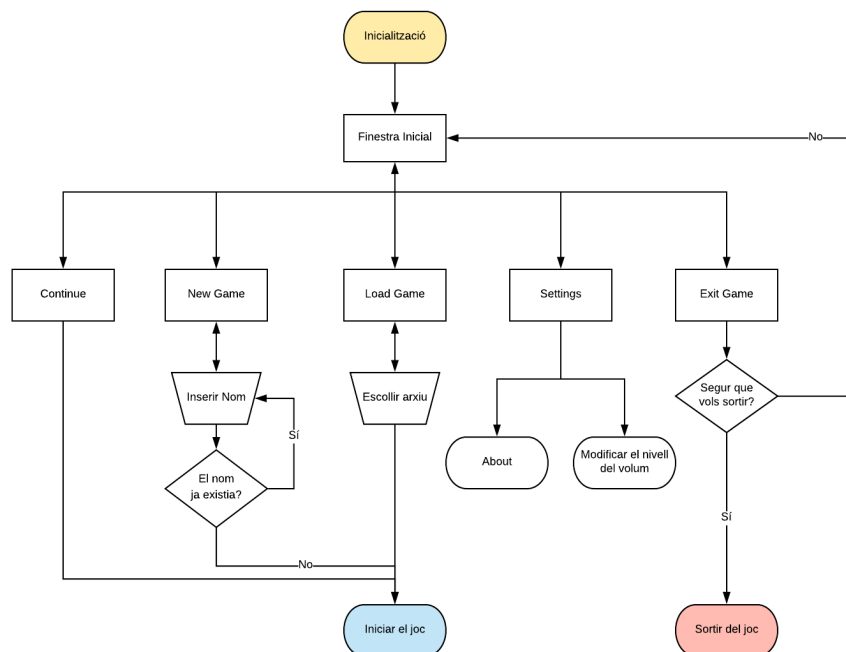


Figura 5.6 Representació de totes les accions que el jugador pot realitzar dins d'aquest menú.

5.6.2. Menú de pausa

El menú de pausa proporciona un accés intuïtiu i ràpid a tots els elements claus del joc. És per tant el menú principal dins del joc. Consta d'una finestra inicial i 5 pestanyes individuals:

- **Inici (Tab Menu):** L'inici és la primera finestra que surt en accedir al menú i permet seleccionar la categoria a què es desitja accedir. Només és possible accedir-hi un cop,

cada vegada que s'obre menú, ja que les pestanyes permeten el desplaçament a qualsevol altra finestra del menú directament. El disseny de la finestra d'inici es pot veure a la Figura 5.7.

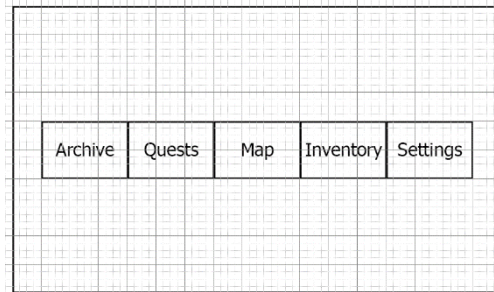


Figura 5.7 Disseny finestra inici (Tab Menu)

- **Arxiu (Archive):** Aquesta pestanya [Figura 5.8 Disseny finestra archive] conté una recopilació de tots els tutorials i de tota la informació educativa que el jugador es troba al llarg de la història. Així mateix, la pestanya arxiu conté petits exercicis interactius que permeten veure de manera molt visual el funcionament dels coneixements adquirits en el joc.

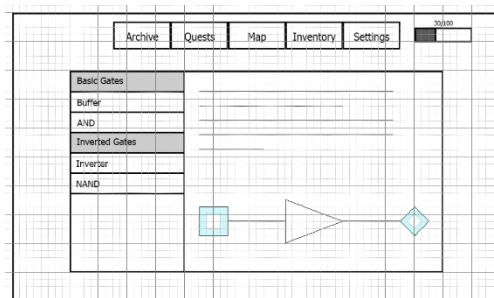


Figura 5.8 Disseny finestra archive

- **Missions (Quests):** L'apartat de missions [Figura 5.9] està format per una llista de les missions actives i completades pel jugador. En seleccionar una missió, se'n mostra una informació detallada i actualitzada del progrés. Així mateix, es pot triar una missió que no estigui finalitzada per seguir-la. Això significa que la informació de la tasca per completar es mostra en la pantalla de joc en tot moment.

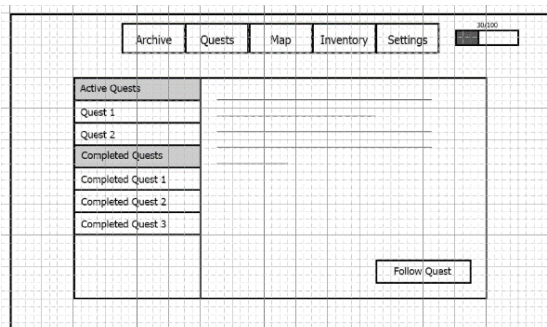


Figura 5.9 Disseny de la finestra quests.

- **Mapa (Map):** El menú del mapa [Figura 5.10] conté una imatge que representa una vista aèria del nivell actual. S'hi representen amb icones aquells elements d'especial importància, així com el jugador. El mapa s'actualitza automàticament en temps real.

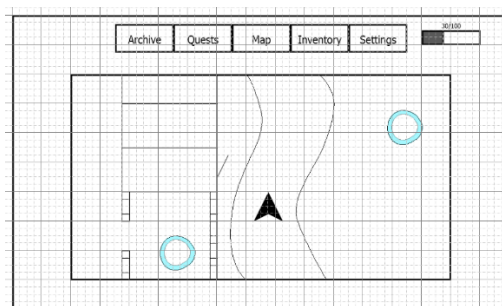


Figura 5.10 Disseny de la finestra mapa.

- **Inventari (Inventory):** Aquí es mostra una llista de tots els objectes que s'han obtingut en el joc, així com la quantitat. El disseny de la finestra de l'inventari es pot veure a la Figura 5.11.

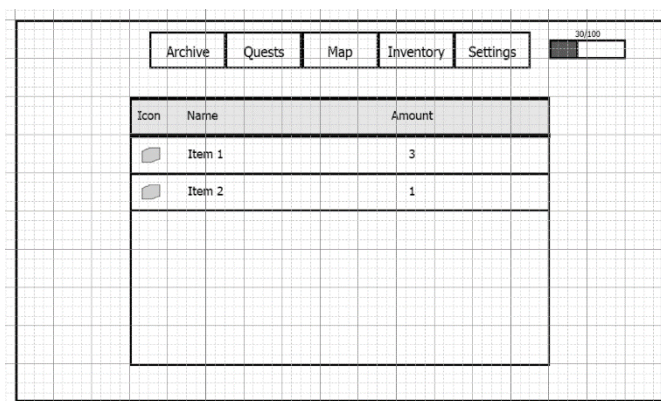


Figura 5.11 Disseny de la finestra Inventory.

Opcions (Settings): Aquesta pestanya [Figura 5.12] permet modificar tant el volum de la

música com el dels efectes de so de manera independent. Així mateix, també permet abandonar la partida i sortir al menú d'inici. Finalment, conté una petita pestanya amb els agraïments i informació sobre els desenvolupadors.

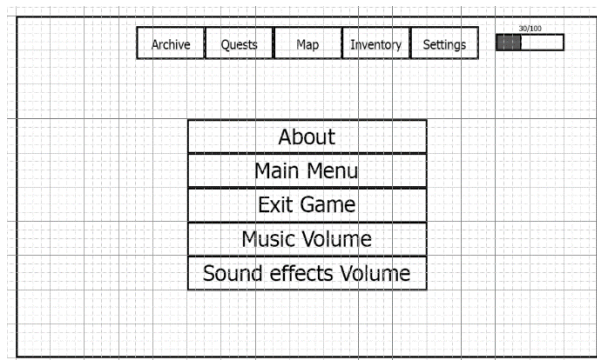


Figura 5.12 Disseny de la finestra settings.

Finalment, la Figura 5.13 presenta un esquema de les possibles transicions dins del menú de pausa. Tal com s'ha dit, gràcies a la zona superior de la finestra, es pot navegar entre totes les finestres indistintament.

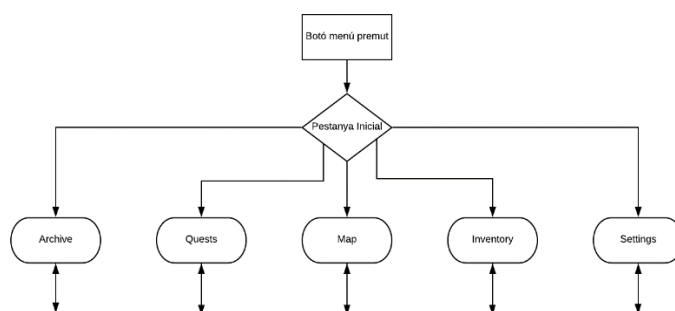


Figura 5.13. Esquema de les possibles transicions dins del menú de pausa.

5.6.3. Heads-Up Display (HUD)

La informació que en tot moment es mostra en pantalla durant la partida, per davant de la pantalla de joc, s'anomena *Heads-Up Display* (d'ara endavant, *HUD*). A continuació, s'exposen els elements més importants i es mostra a la Figura 5.14 un exemple de totes les finestres informatives que poden aparèixer en la pantalla de joc, tot i que mai hi seran totes al mateix temps:

- **Minimapa:** Vista aèria del mapa en temps real. Permet facilitar l'orientació i la cerca de punts d'interès. Sempre és visible a la cantonada superior dreta de la pantalla.
- **Missió activa:** Títol i descripció de la missió actualment activa. Proporciona al jugador informació ràpida del seu objectiu. Sempre és visible a la cantonada superior dreta de la pantalla, per sota del minimapa.
- **Tooltips:** Els *tooltips* són finestres emergents que apareixen en succeir certs esdeveniments i donen alguna informació útil al jugador. Alguns exemples són: recollir objectes, avançar en la missió o activar un objecte. Aquestes finestres desapareixen quan passa un temps o si el personatge s'allunya del punt d'interès.

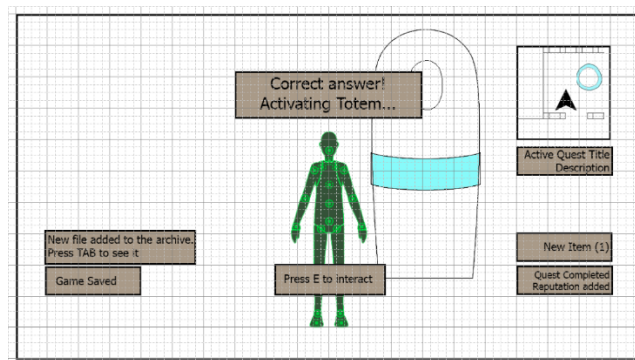


Figura 5.14. Exemple de totes les finestres informatives que poden aparèixer en la pantalla de joc.

6. Decisions de disseny

6.1. Motor de videojocs

Fins ara s'ha comentat quin és el joc a desenvolupar i totes les seves característiques. Un cop fet això, s'ha de decidir com s'assolirà aquest objectiu i quines eines s'utilitzaran. Per al desenvolupament de videojocs i aplicacions existeixen diversos programaris anomenats motors de videojocs, dels quals, els més coneguts arreu del món són *Unity3D* i *Unreal Engine*. Un motor de videojocs és un programa que permet el disseny i representació d'un videojoc per a diferents sistemes operatius i té diverses característiques específiques, com un motor gràfic que és capaç de fer la renderització de models gràfics en 3D. També s'ha considerat l'opció de crear el videojoc mitjançant el llenguatge de programació *Python*. A continuació s'exposen les diferents característiques d'aquestes alternatives i s'argumenta la decisió de l'eina triada.

Unity3D és un motor de videojocs multiplataforma amb suport per a desenvolupament de videojocs en 2D i 3D [30]. L'edició personal és gratuïta i conté molta documentació d'ajuda al desenvolupament i una comunitat activa disposada a respondre dubtes. Es pot triar entre tres llenguatges de programació: *C#*, *JS* i *Boo*. A més, es pot gaudir de la *Asset Store*, una botiga virtual on es poden descarregar diversos components i programaris útils.

De la mateixa manera, *Unreal Engine* és un motor de videojocs multiplataforma que permet desenvolupar jocs en 2D i 3D i és gratuït per a ús personal [31]. A més, també conté documentació, una comunitat activa i una botiga virtual amb un funcionament similar al de *Unity3D*. Les diferències principals són alguns elements de funcionament intern del programa i el llenguatge de programació que s'utilitza, el *C++*.

Pel que fa a *Python*, com que es tracta d'un llenguatge més senzill que s'ha estudiat amb molta profunditat a la universitat, s'han buscat eines de disseny de videojocs en 3D o la manera de crear-les. Només s'ha trobat el motor de videojocs *Panda3D*. Malauradament, disposa de menys funcionalitats que les altres alternatives i per tant s'ha descartat [32].

En definitiva, *Unity3D* i *Unreal Engine* tenen característiques molt similars, però la primera eina té una corba d'aprenentatge més senzilla pel fet que *C#* és més fàcil d'aprendre que *C++*. Això pot alliberar més temps per a aprendre altres funcionalitats del motor i així poder crear el joc de manera més efectiva.

6.2. Recursos artístics

Durant el transcurs del projecte s'han fet servir diverses fonts per obtenir recursos gràfics, així com eines pròpies de disseny.

Tots els gràfics que s'han obtingut d'Internet han sigut sempre de llicència oberta. Principalment s'han fet servir les següents utilitats:

- **Unity Asset Store:** Lloc web integrat amb *Unity* que permet als desenvolupadors compartir recursos de diversa índole. Els recursos pel disseny d'escenaris i objectes s'han obtingut principalment d'aquí. [14]
- **Mixamo:** Aquesta pàgina permet descarregar de manera gratuïta una selecció de personatges i animacions aptes per a ser utilitzades a *Unity*. [15]
- **OpenGameArt:** Pàgina web que disposa d'una gran quantitat de recursos artístics de lliure disposició. En aquesta pàgina destaquen especialment imatges i efectes de so [16].

Per a l'edició d'aquell material d'Internet que ho requerís, així com per a la creació de nous recursos, va ser necessària la utilització de diverses eines de disseny. A continuació, es llisten les més destacades:

- **Photoshop:** Editor d'imatges de mapa de bits. S'ha utilitzat per a la realització de models en dues dimensions, la creació de tota la interfície i l'edició de textures en tres dimensions. [33]
- **Inkscape:** Editor de gràfics vectorials que s'ha utilitzat per primer cop a la universitat a l'assignatura *Teoria de màquines i mecanismes*. Ha permès realitzar els esquemes vectorials de tots els tests d'electrònica. [34]

6.3. Recursos d'organització

Una bona organització d'un projecte requereix eines que ajudin a estalviar temps i a augmentar la productivitat del grup. Hi ha molts tipus d'eines que serveixen per diferents propòsits; algunes serveixen per transferir fitxers de manera eficient entre els membres del grup, altres són útils per a la planificació de projectes. A continuació, s'exposen les eines destinades a l'organització del projecte que s'han emprat en el desenvolupament d'aquest projecte:

6.3.1. Trello

Trello és una eina col·laborativa que organitza els projectes en una pantalla que simula una pissarra o una agenda amb targetes que s'hi enganxen [35]. A cada targeta no només se li pot introduir text, sinó que se li poden adjuntar enllaços, llistes, imatges, escriure comentaris i assignar dates límit, entre d'altres. També es poden associar significats a diferents colors associats a cada targeta. Per exemple, es pot assignar el color verd a “*prioritat baixa*”, el color groc a “*prioritat mitjana*” i el vermell a “*prioritat alta*”. A més, a l'historial de canvis es pot veure tot el que ha realitzat cada membre del grup, gestionar els membres del grup i dur a terme una cerca d'alguns elements específics de la pissarra. Gràcies a totes aquestes característiques, es pot veure a primera vista tota la informació relacionada amb un projecte. La Figura 6.1 mostra la pissarra que s'ha fet servir en el desenvolupament d'aquest projecte i serveix d'exemple de com pot ser una pissarra a *Trello*.

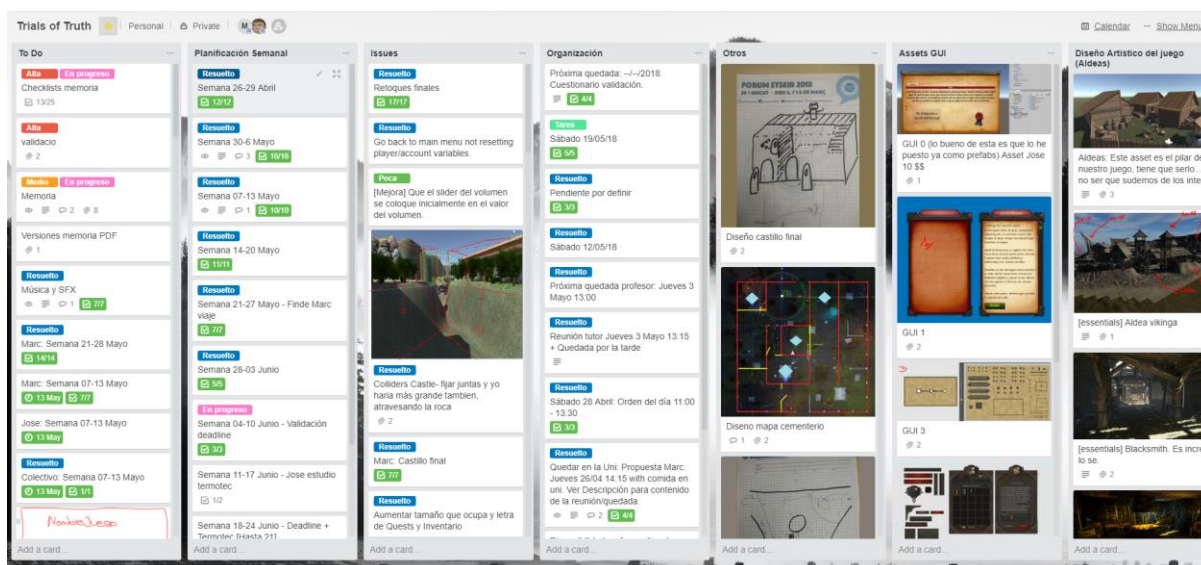


Figura 6.1 Imatge de la eina web Trello.

La primera llista de l'esquerra mostra les tasques més importants a realitzar i tenen una prioritat assignada segons el color de la barra superior de cada targeta. A més, sempre s'ha afegit una targeta amb una llista de tasques de la setmana actual per a cada membre del grup, de tal manera que cada membre del grup pot veure la seva feina i la del seu company de manera ràpida i eficaç. Totes les tasques ja resoltes es marquen de color blau com a “Resolt” en comptes d'eliminar-les, per així poder consultar aquesta informació en qualsevol moment.

A la segona columna, es mostra la planificació setmanal del projecte, amb tots els objectius que s'han assignat amb antelació. Les tasques dins de la targeta associada a una setmana

no són específiques i serveixen per a assignar tota la feina a cada membre del grup amb més exactitud en la seva respectiva setmana.

Una altra categoria important és la de *Issues*, ja que en aquesta es documenten tots els problemes que sorgeixen durant el desenvolupament del joc, siguin errors de codi, de disseny o fins i tot dubtes ocasionals. Així, cap problema no passarà desapercebut. A més, quan algú troba un error relacionat amb la feina del company, la pot posar a aquesta llista perquè tingui aquesta informació quan revisi la pissarra.

El menú lateral de la dreta de la pantalla mostra l'historial d'activitat dels membres del grup i diverses accions que es poden realitzar sobre el projecte. Finalment, s'han afegit moltes altres llistes per agrupar algunes eines que poden servir d'ajuda durant el projecte i informació addicional de menor rellevància.

6.3.2. Google Drive:

Google Drive permet compartir fitxers entre diferents usuaris, com moltes altres eines [36]. Però el seu punt fort resideix en el fet que es poden editar documents de text, de presentació o fulls de càlcul en temps real, amb diferents persones. Gràcies a això, s'ha pogut treballar de manera més eficaç en la memòria del projecte i en la presentació de *PowerPoint*.

6.3.3. Unity Collaboration:

Tal com s'ha esmentat en les decisions de disseny per al motor de videojocs, *Unity* ofereix un servei gratuït per desar tota la informació d'un projecte al núvol de fins a 1 GB d'espai [37]. Això permet a ambdós membres del grup treballar en el mateix projecte al mateix temps i desar els canvis al núvol; llavors l'altre membre rep la notificació de descarregar els nous canvis. A condició que no es treballi exactament el mateix àmbit del projecte, mai hi ha problemes amb el fet de treballar a la vegada.

El projecte d'aquest videojoc ha acabat ocupant un espai d'aproximadament 5 GB. Per aquesta raó, s'ha hagut d'optar per una subscripció mensual de 5 € durant quatre mesos per tenir fins a 26Gb d'espai disponible al núvol.

7. Introducció a *Unity*

Unity és un programa desenvolupat per *Unity Technologies* orientat al desenvolupament de videojocs per a ordinador, consoles, dispositius mòbils i navegadors d'internet. Es tracta d'un software amb moltes opcions i funcionalitats. Tot i que és de gran complexitat, disposa d'una bona documentació que s'ofereix a la pàgina web. A continuació, s'expliquen els conceptes més importants que calen per entendre el funcionament de *Unity*.

7.1. Assets

Un *asset* és una representació d'un objecte que es pot fer servir en un projecte. Pot venir d'un fitxer creat fora de *Unity*, com una imatge, un fitxer d'àudio o un model 3D, però també pot venir de *Unity*. A més, la *Unity Asset Store* [14] ofereix una àmplia selecció de paquets d'*assets* de diferents tipus, tant de pagament com gratuïts. Aquests paquets són recopilacions de fitxers que es poden afegir a un projecte un cop adquirits.

7.2. Scenes

Les *Scenes* (escenes) contenen l'entorn i els menús del joc. Comunament, en un projecte, cada *scene* conté un nivell del joc per separat.

7.3. GameObject

Els *GameObject* són el concepte més important de *Unity*. Cada objecte del joc és un *GameObject*, des de personatges, llums, càmeres, fins a efectes especials. Però un *GameObject* no pot fer res per si sol; se li han d'afegir propietats abans que pugui ser un personatge, un entorn o un efecte especial. Perquè tingui les propietats desitjades se li han d'afegir *Components*. Depenent de l'objecte que es vulgui crear, la combinació de components que se li han d'assignar és diferent. A Figura 7.1 es pot veure un esquema del funcionament d'un *GameObject* amb els seus components.

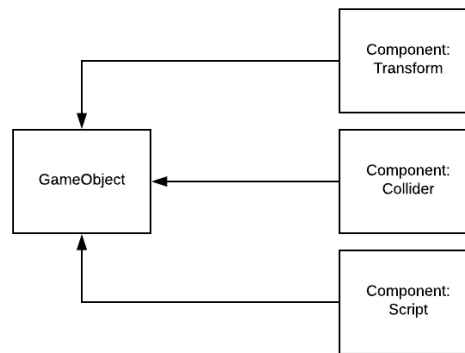


Figura 7.1 esquema del funcionament d'un *GameObject* amb els seus components

A més, un *GameObject* pot contenir altres *GameObjects*, com si fossin carpetes que dins contenen diferents fitxers. Un *GameObject* que conté altres *GameObjects* és el seu *parent* i els seus continguts, els seus *children*.

7.4. Components

Afegeixen funcionalitats als *GameObjects* als quals pertanyen. El component més comú, que pertany a tots els *GameObjects*, és el *Transform*. Aquest component defineix la posició, rotació i escala de l'objecte a la *Scene*. Un *GameObject* molt important — ja que se'n requereix almenys un a cada *Scene* — és el *Main Camera*. Es tracta de la càmera principal del joc, la qual es fa servir per seguir al personatge en tot moment i conté, per defecte, alguns components que li atorguen les funcionalitats, com la *Camera* (per gravar i mostrar la *Scene*) i el *Audio Listener* (per reproduir el so i la música). Alguns dels components més utilitzats en el projecte són els següents:

- **Collider.** És una forma geomètrica, com un cub en el cas d'un *Box Collider* o una esfera per a un *Sphere Collider*. Gestiona les col·lisions amb altres *Colliders* de l'entorn, de manera que no es travessin. Se li pot afegir la funcionalitat *Trigger* per gestionar col·lisions sense intervenir en el moviment i col·lisions físiques de l'element amb què xoca.
- **CharacterController.** Es tracta d'un component que afegeix un sistema de càlculs i d'aplicació de lleis físiques bàsiques al personatge, mitjançant un tipus de *Collider* (*CapsuleCollider*). Afegeix funcionalitats de moviment, com és ara, l'angle màxim que pot pujar el *GameObject* associat en un pla inclinat.
- **AnimatorController.** Gestiona les animacions del personatge segons una màquina d'estats i animacions associades a cada estat.
- **Canvas.** Per a *GameObjects* de la *UI*, substitueix el component *Transform* i fa una

tasca similar amb aquests objectes.

- **Altres elements d'UI** com per exemple *InputText*, *Button*, *Image* que permeten a l'usuari escriure en camps de text, prémer botons i veure imatges.
- **Script**: Es pot associar un programa creat pels creadors del joc a un *GameObject* de la mateixa manera que els components, per realitzar diferents accions d'interès.

7.5. Prefabs

Quan s'ha creat un *GameObject* amb els components que interessen i s'ha ajustat a gust del programador, es pot desar com a *Prefab*, el qual pot ser reutilitzat sense haver de tornar a associar cada component i els valors de les seves variables cada cop. Així doncs, els *Prefabs* són, essencialment, *GameObjects* prefabricats.

7.6. Editor de Unity

A Figura 7.2 es mostra un exemple de l'editor de *Unity* amb una identificació dels elements més importants, dels quals es dóna una breu explicació a continuació:

- **Scene View**: Permet visualitzar i navegar per la *Scene*. Es pot veure en *2D* o *3D* i en diverses perspectives. S'utilitza per editar la *Scene* i crear els nivells.
- **Hierarchy**: Presenta en una llista jeràrquica tots els elements d'una escena.
- **Inspector**: Aquesta finestra permet veure i editar totes les propietats de l'objecte seleccionat actualment.
- **Toolbar**: Agrupa l'accés a moltes característiques de l'editor. A l'esquerra hi ha eines bàsiques per manipular la *Scene*, al centre els botons de *Play*, per provar el joc des de l'editor. Els botons de la dreta donen accés a altres eines d'interès.
- **Game View**: Mostra el que es veu en el joc des de la perspectiva de la càmera. Si s'ha fet *Play* des de la *toolbar*, es pot jugar al joc per fer proves sense necessitat de compilar-lo en un fitxer executable.

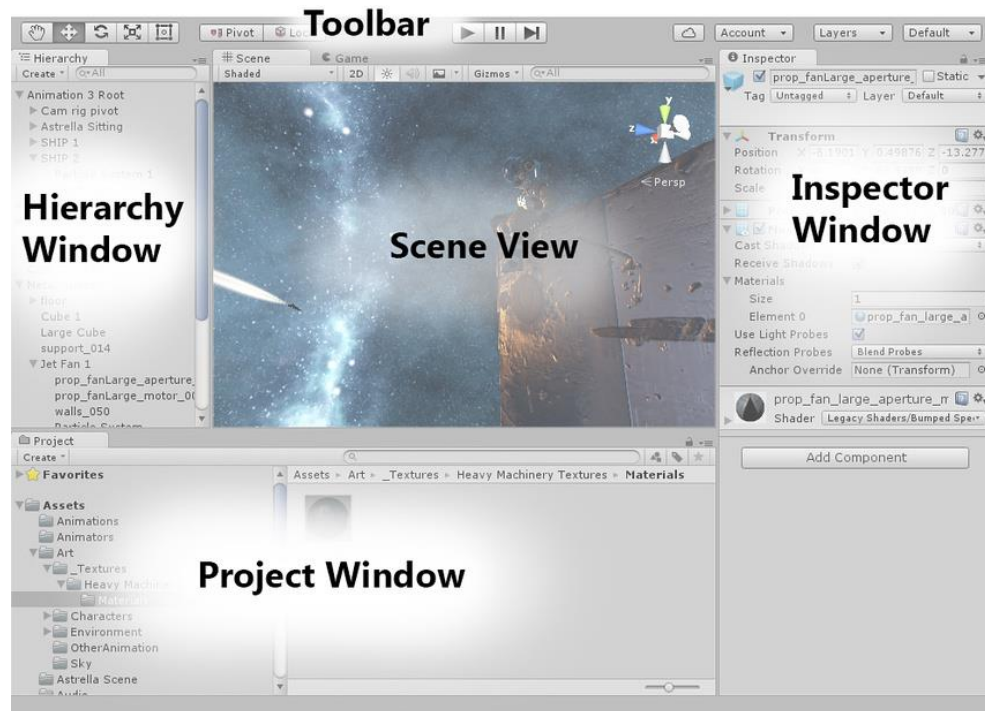


Figura 7.2 editor de Unity amb una identificació dels elements més importants

7.7. Editor de Unity

Generalment, la idea d'un programa tradicional consta d'un bucle principal on s'executa el codi fins que s'acaba. En comptes d'això, *Unity* passa el control als *scripts* de manera intermitent, mitjançant crides a determinats mètodes que posseeix cada *script*. Un cop s'ha acabat l'execució de la funció, es retorna el control a *Unity*. Aquestes funcions s'anomenen *funcions d'esdeveniment* perquè s'activen en resposta a esdeveniments que succeeixen durant l'execució. S'utilitza una nomenclatura per a les funcions segons l'esdeveniment al qual responen. Per exemple, la funció *Update* es duu a terme just abans de l'actualització d'un *frame*, i la funció *Start* s'executa just abans de la primera crida d'*Update* d'un objecte. Hi ha moltes altres funcions d'esdeveniment que seran emprades al llarg del projecte, les quals s'explicaran més endavant. Cal mencionar que *Unity* crida totes aquestes funcions a cada *frame*, algunes més d'un cop. Un esquema més detallat d'aquestes crides es pot veure en la Figura 7.3.[38]

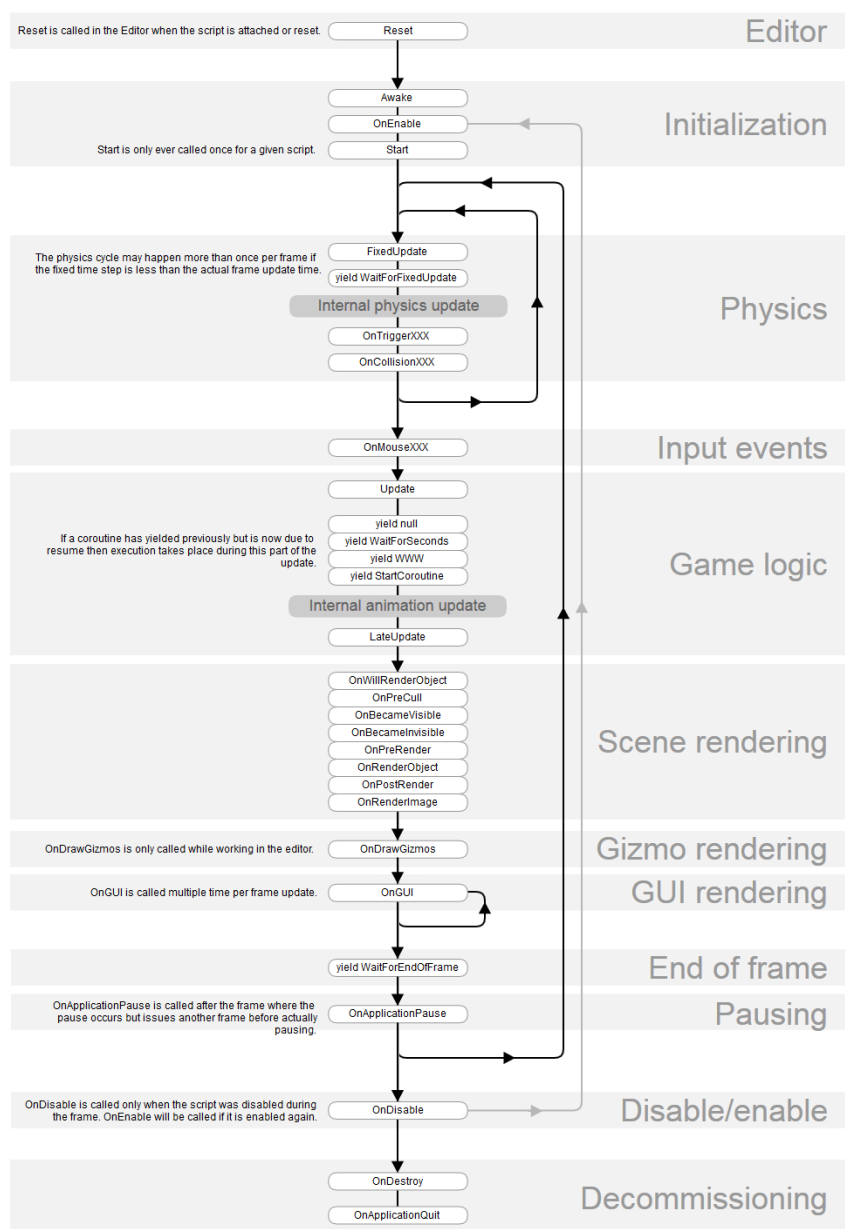


Figura 7.3 Esquema de les funcions d'esdeveniment.

8. Implementació del disseny

Un cop s'han après les bases de *Unity* és el moment de fer la implementació del joc. Es pot dividir en moltes categories, les quals s'exposen a continuació.

8.1. Pantalla de joc

La pantalla de joc conté no només l'entorn i el personatge, sinó que també hi ha informació addicional superposada amb elements del *HUD*. A continuació, es mostra un exemple de com es veu la pantalla [Figura 8.1], i s'expliquen tots els elements que hi apareixen i d'altres que només són visibles en algunes ocasions.



Figura 8.1. Exemple de com es veu la pantalla

En primer lloc, el *minimapa* es mostra a la cantonada superior dreta de la pantalla i mostra una imatge de l'entorn vista des de dalt i que s'actualitza a cada instant. El que grava la càmera es converteix en temps real en una *textura*, la qual és llegida com una imatge. Tots els *GameObjects* per renderitzar s'han categoritzat en una *layout*. Una *layout* a *Unity* ordena els objectes en diferents categories com terreny, fons de pantalla, objectes sense interaccions físiques, etc., de manera que alguns *scripts* poden seleccionar les categories d'objectes que poden dur a terme determinades accions. Per exemple, en aquest cas la càmera del minimapa renderitza només els objectes que són a la *layout minimapa*.

En segon lloc, podem veure la finestra de l'objectiu de missió actual a sota del minimapa. Al menú de pausa es pot triar la missió de la qual es vol rebre indicacions per mostrar-les en la finestra. El text de la descripció de cada missió es desa en una variable global i el mètode *Update* de la finestra actualitza el text de la missió.

A part de les finestres anteriors, n'hi ha d'altres que apareixen en diferents ocasions, anomenades *tooltips*. Per exemple, a la part inferior de la pantalla de la Figura 8.1 anterior

surt una petita finestra que indica que es pot prémer la tecla de la lletra *E* per interactuar amb el *tòtem*. Existeixen dues altres finestres similars per interactuar amb altres objectes. Totes elles apareixen quan el jugador pot realitzar una acció especial com interactuar amb un *tòtem*, amb un *NPC* o desar l'estat de la seva partida. Generalment, els *tooltips* són a la pantalla, però el *GameObject* es troba en estat desactivat i no es renderitza. Mitjançant la crida *GameObject.SetActive*, es pot canviar aquest estat per mostrar l'objecte en pantalla quan sigui rellevant i amagar-lo quan deixi de ser-ho.

Quan s'agafa un objecte del terra, poden mostrar-se fins a dos altres *tooltips* a la pantalla, amb més informació [Figura 8.2]. A la part inferior dreta sempre s'indica durant uns segons el nom de l'objecte que s'ha agafat i la seva quantitat entre parèntesis. El *tooltip* de l'esquerra s'activa si l'objecte compleix la característica de fer disponibles més entrades de tutorial a l'*Archive*. Es compta el temps transcorregut a cada *frame* i en el moment en què es sobrepassa un temps límit, es desactiven les finestres. Amb el mateix concepte funcionen les finestres que informen de quan la partida s'ha desat satisfactòriament i la finestra que informa de si el resultat dels tests d'un *tòtem* han estat correctes o no.

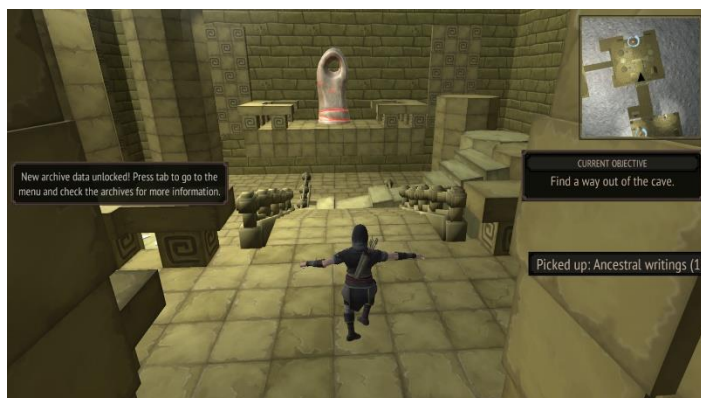


Figura 8.2 Tooltips a la pantalla amb més informació

Finalment, existeixen les finestres del tutorial que es mostren en diferents situacions durant el primer nivell. Els tutorials mostren consells bàsics referents als controls del personatge i el funcionament dels tests i menús; apareixen quan el jugador arriba a un punt de l'entorn o compleix certes condicions (com l'estat d'una missió). Tenen una estètica similar a les finestres de tests i ocupen el centre de la pantalla, tal com es pot veure en l'exemple de la Figura 8.3.

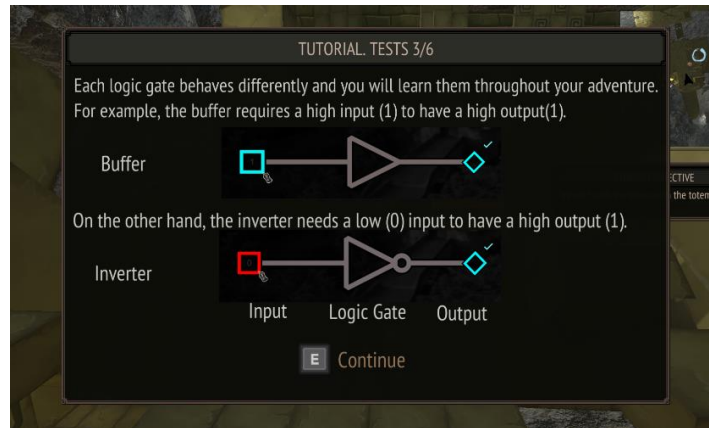


Figura 8.3 finestres del tutorial

8.2. Personatge Principal

La creació del personatge principal està compromesa per molts aspectes, com la tria del seu model 3D, les animacions, la programació del moviment i diferents interaccions del personatge i la càmera que el segueix. Tot seguit, s'explica tot el procediment que s'ha dut a terme per al seu desenvolupament.

8.2.1. Model del personatge

El personatge i les seves animacions s'han obtingut principalment de la pàgina *Mixamo.com*, la qual distribueix una àmplia selecció de personatges i animacions gratuïtament. Permet la descàrrega d'un fitxer de format *FBX* que es pot importar a l'editor de *Unity*. Tot i que s'ofereixen molts tipus diferents de personatges, se n'ha de triar un que encaixi amb el disseny de l'entorn i l'estil artístic. El personatge principal és, doncs, el que es mostra a la Figura 8.4. Un cop s'ha triat el model, s'afegeix el personatge a una *Scene* de *Unity* i es pot començar a crear i a afegir components necessaris perquè funcioni com s'ha previst.



Figura 8.4 Personatge principal del videojoc.

A la Figura 8.5 es veu el personatge a la *Hierarchy* quan s'ha afegit a una *Scene*. Com es pot observar, el *GameObject* *erika_archer* conté altres *GameObjects* que a la vegada tenen diferents aspectes del model del personatge. S'afegiran diversos components a l'objecte superior de la jerarquia per tal d'atorgar-li funcionalitats d'un personatge controlable per al jugador.

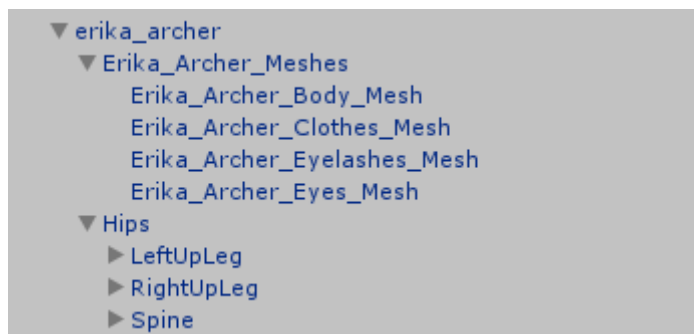


Figura 8.5 Vista del personatge a la *Hierarchy* a l'editor de Unity.

8.2.2. Character Controller

Unity té un component anomenat *CharacterController* que és de gran utilitat per dissenyar un personatge i després afegir tota la lògica del seu moviment en un *Script* apart, que també s'afegirà com a component al mateix *GameObject*. A més, incorpora un *Capsule Collider* per al càlcul de col·lisions i moviments. Una altra utilitat que té aquest component és la informació que es pot obtenir mitjançant algunes variables, com *isGrounded*, que és un booleà que diu si el *Collider* es troba en contacte amb una superfície a sota. A la Figura 8.6 es pot veure l'aspecte del *Capsule Collider* en el personatge a la *Scene View* de l'editor i les opcions que ofereix el *CharacterController*.

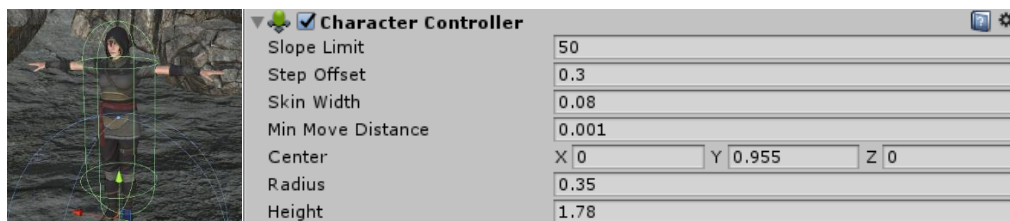


Figura 8.6 *CapsuleCollider* i les seves opcions a l'editor de Unity.

A continuació, es descriuen breument les diferents característiques del *CharacterController*:

- **Center, Radius i Height.** Defineixen el *Collider* del personatge. La posició és relativa al *GameObject* a què està associat.

- **Slope Limit:** L'angle màxim que pot pujar el personatge en un pla inclinat. En aquest cas, no pot caminar per angles superiors a 50 graus.
- **Step Offset:** El personatge pot pujar escales només si aquestes tenen una alçada inferior a aquest nombre (en metres).
- **Skin Width:** Dos *Colliders* poden travessar-se amb una profunditat tan gran com el seu *Skin Width*.
- **Min Move Distance:** El personatge només es mourà si la seva velocitat és superior al nombre indicat.

8.2.3. Màquina d'estats

Una màquina d'estats és una sèrie d'estats i transicions d'estats subjectes a diferents condicions. Per gestionar les animacions s'ha fet servir un altre component, anomenat *Animator*, al qual se li ha d'afegir un *Animator Controller*, que funciona com una màquina d'estats, en la qual a cada estat hi ha associada una animació o una sèrie d'animació. Sempre que el personatge es trobi en un estat, es mourà segons l'animació corresponent.

Dins d'una màquina d'estats pot haver-hi els següents elements:

- **Estats:** Un estat que té associada una animació.
- **Fletxes:** Indiquen les possibles transicions entre estats, sempre que es compleixin certes condicions configurables.
- **Altres màquines d'estats:** Per organitzar millor l'esquema d'una màquina d'estats, es poden posar estats que en realitat són altres màquines. Així, es poden categoritzar diferents animacions del personatge.
- **Blend Trees:** Múltiples animacions que poden combinar-se segons el valor d'un o dos paràmetres, generalment nombres decimals.
- **Any State:** Serveix per tenir transicions a un estat al qual es pot arribar des de qualsevol altre estat, en comptes de tenir una fletxa des de cada estat cap al destí.
- **Entry:** Defineix l'estat inicial dins de la màquina d'estats.
- **Exit:** Defineix la sortida d'una màquina d'estats a l'estat superior.

Per al cas del personatge principal, s'ha distingit inicialment si el personatge està enlairat (*Airborne*) o si està a terra (*Ground Movement*) [Figura 8.7]. Ambdós elements són màquines d'estats i contenen altres estats, tal com s'ha explicat anteriorment. Començant pel moviment a terra, l'estat inicial és *Free Movement*, i des d'aquest estat només es pot tornar a la capa superior per, per exemple, saltar; el personatge en aquest cas estarà enlairat i haurà de transicionar a un altre estat. *Free Movement* és un *Blend Tree* que conté quatre possibles animacions segons la velocitat de moviment del personatge [Figura 8.8]:

- **Idle:** El personatge està quiet si la seva velocitat de moviment és nul·la.
- **Walk Forward:** El personatge camina cap endavant a poc a poc.
- **Run Forward:** El personatge corre cap endavant.
- **Sprint Forward:** El personatge esprinta cap endavant.

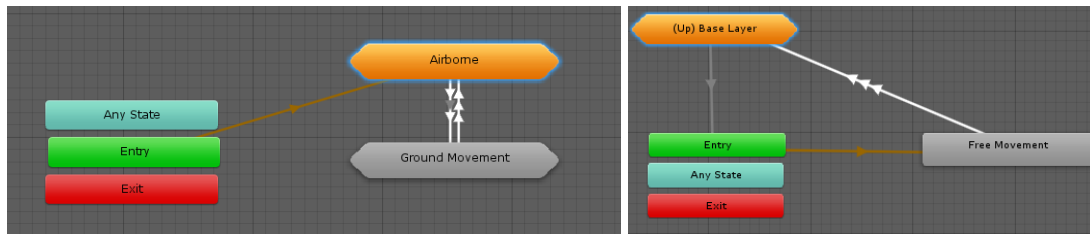


Figura 8.7 Màquina d'estats del personatge principal.

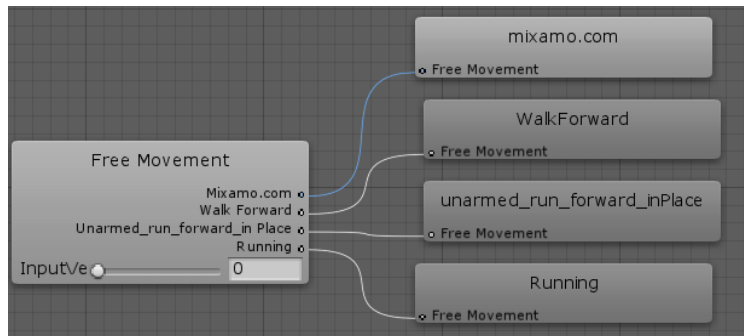


Figura 8.8 Blend Tree de l'estat Free Movement.

Si el jugador saltés o caigués per un precipici, l'estat variaria cap a la màquina *Airborne* que es mostra a la Figura 8.9. El flux de transicions passa per una animació de saltar des del terra, una d'estar enlairat caient, i una d'aterrar i amortir la caiguda.

Les condicions que regeixen si s'ha de canviar d'un estat a un altre es defineixen a l'*AnimatorController*. Aquestes condicions, que poden ser editades per uns altres *Scripts*, són les següents:

- **InputVertical:** Defineix mitjançant un decimal que pot valer entre 0 i 1 la velocitat del personatge respecte a la seva velocitat màxima de moviment. Si està esprintant, se suma 0,5 al valor i pot valer fins a 1,5. És el paràmetre més important en el moment de definir les animacions del moviment a terra.
- **IsGrounded:** És un booleà que val *True* si el personatge és a terra i *False* si no hi és.
- **IsSprinting:** És un booleà que val *True* si el personatge està esprintant i *False* si no ho està.
- **Jump:** És un booleà que val *True* si el personatge ha saltat i *False* si no ho ha fet.

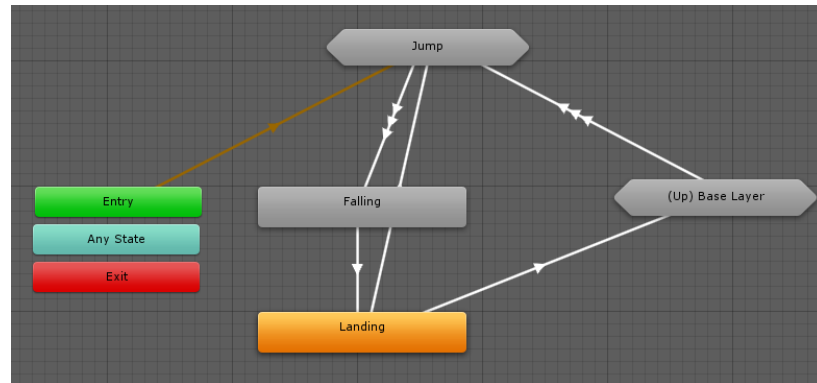


Figura 8.9 Màquina d'estats del personatge per a quan està enlairat.

8.2.4. Player Controller

Un cop s'ha dissenyat la màquina d'estats amb totes les animacions afegides i adaptades correctament i s'ha afegit el *Character Controller* al *GameObject* del personatge principal, cal crear un *Script* que permeti a l'usuari controlar el personatge. Aquest *Script* donarà la informació necessària a la màquina d'estats per mostrar les animacions correctes, a partir de les accions que realitzi el jugador. El *Script* accedirà als altres components del *GameObject* després d'inicialitzar-los a una variable, mitjançant la crida *GetComponent<Classe>()* i modificarà alguns valors, com per exemple els paràmetres que s'han mencionat anteriorment, que serveixen per fer les transicions entre estats de les animacions del personatge.

En inicialitzar la *Scene*, el primer que es crida és el mètode *Start()*, que s'encarrega d'inicialitzar algunes variables. A continuació, es crida a *FixedUpdate()*, en la qual es comprova quins botons ha premut el jugador i s'actualitzen els paràmetres de l'*Animator*. Tot seguit, *Unity* crida la funció *Update()*, en la qual es realitzen la majoria de les accions del *Script*. Per una banda, mitjançant *MoveCharacter()*, *Sprint()* i *Jump()* el personatge es mourà segons els valors que s'han obtingut anteriorment quan s'han comprovat les accions del jugador. Aquests tres mètodes gestionen la velocitat de moviment i la rotació del personatge en totes les direccions i el moviment a causa de la gravetat. Per altra banda, s'ha creat un altre mètode que es crida durant l'*Update()* anomenat *CheckGroundDistance()*, que s'encarrega de comprovar la distància dels peus del personatge a terra i, en cas que aquesta sigui inferior a un valor concret i que no s'hagi saltat, mou el personatge directament a terra, de manera que els peus queden enganxats a la superfície. Això es fa perquè en superfícies irregulars i descendents el personatge semblava que rebotés en avançar. Finalment, el mètode *SlopeDirection()* comprova l'angle que fa la superfície del terra que està trepitjant el personatge respecte de l'eix horitzontal. Si aquest angle és molt vertical, el personatge llisca pel pla inclinat i cau, de manera que s'evita que es puguin pujar plans molt verticals.

PlayerController també incorpora altres mètodes i variables interessants, com la gestió del menú, la recollida d'objectes del terra, l'actualització de l'estat de missions i la mostra de missatges del tutorial. Aquests aspectes s'expliquen més endavant en els apartats que tenen més relació amb ells. Amb tot, a la Figura 8.10 es pot veure com es presenta el component a l'editor de *Unity*, que mostra en cada component primer el nom del *Script* i seguidament el nom de totes les variables d'accés *public*.

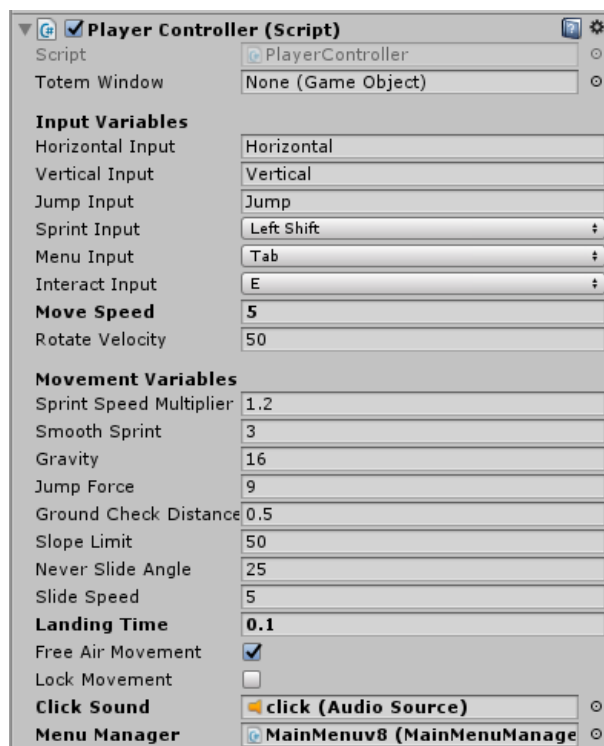


Figura 8.10. Vista del component *PlayerController* a l'editor de *Unity*.

8.2.5. Càmera

La càmera és el dispositiu que captura i ensenya el món al jugador. També segueix el personatge principal a una certa distància i es pot rotar lliurement amb el ratolí. A la Figura 8.11 es pot veure una imatge de la visió de la càmera en el joc i la seva representació com a objecte extern a l'escena.



Figura 8.11. Visió de la càmera dins del joc i representació de la càmera a l'escena respectivament.

El control de la càmera es realitza mitjançant dos *Scripts*. El primer és el *Script* anomenat *FollowPlayer*, que fixa la distància de la càmera respecte al personatge. A més a més, s'encarrega de llegir la variació vertical i horitzontal del ratolí i de transformar aquesta mesura en rotació al voltant del personatge. Així mateix, *FollowPlayer* afegeix restriccions a la rotació de la càmera per sobre i per sota del personatge. Aquesta funció s'inclou en gairebé la totalitat de jocs per evitar un control antiintuïtiu. A la Figura 8.12 es pot veure la rotació màxima de la càmera respecte a la direcció normal de la imatge.



Figura 8.12. Rotació màxima de la càmera respecte a la direcció normal a la imatge.

El segon *Script*, anomenat *CameraCollision*, gestiona les col·lisions en espais confinats. Aquest codi s'activa quan la càmera entra en contacte amb el terra o amb un objecte i s'apropa al jugador dinàmicament. D'aquesta manera s'evita que la càmera es quedi travada o passi a través dels objectes. A la Figura 8.13 es pot veure un exemple de la càmera quan col·lideix amb una paret.



Figura 8.13. Càmera en col·lisió amb una paret.

8.3. Interaccions amb l'entorn i missions

En el disseny conceptual del videojoc s'ha exposat el funcionament dels elements mitjançant els quals pot interactuar el personatge principal. Aquests són els *objectes*, *tòtems* i *NPC*. Ara es mostrarà la implementació de tots aquests elements en *Unity* i com es lliguen aquests elements amb les missions principals i secundàries.

8.3.1. Missions

Les missions són útils en el joc perquè donen al jugador diferents tasques a realitzar i la possibilitat de triar quines vol fer i en quin ordre. Això permet tenir llibertat per explorar el món alhora que es duen a terme les tasques. Com s'ha explicat durant el disseny conceptual, hi ha dos tipus de missions, les principals i les secundàries. En l'àmbit de funcionament no difereixen gaire, així que s'explicaran de forma conjunta. Durant una missió, el jugador pot trobar-se en diferents estats. Un estat, per exemple, seria quan acaba de començar la missió i un altre és quan la finalitza, però hi pot haver més estats. Per això, cada estat s'identifica mitjançant un nombre enter, on el zero representa l'estat inicial. Per altra banda, cada estat d'una missió pot tenir associat un text descriptiu que pot ajudar el jugador a saber quina és la seva tasca. Un exemple d'una missió segons els estats seria el següent:

- **Estat 0:** Acaba de començar la missió i ha de dirigir-se cap al poble.
- **Estat 1:** El jugador ja ha arribat al poble i ha d'obtenir l'objecte que se li ha demanat.
- **Estat 2:** L'objecte s'ha obtingut, s'ha de retornar al *NPC* per completar la missió.
- **Estat 3:** S'ha retornat al *NPC*. Missió completada.

La classe *PlayerController* conté una variable de tipus *estàtica*, a la qual, per tant, es pot accedir des de qualsevol *Script* sense necessitat de cridar una instància d'un objecte de *PlayerController*, anomenada *quests*. És un *Dictionary<string, int>* que conté el nom de cada missió i l'estat en el qual es troba el jugador per a cadascuna d'elles. Així doncs, diversos mètodes estàtics de *PlayerController* s'encarreguen de llegir i actualitzar l'estat de les missions:

- **GetQuestState:** Retorna l'estat de la missió o -1 en cas que no s'hagi començat.
- **StartQuest:** Comença una missió, tot posant-la a l'estat inicial.
- **SetQuestState:** Actualitza l'estat d'una missió i, en cas d'haver-la finalitzat, en comença una de nova de les que el jugador tingui començades però no finalitzades.

L'estat del jugador en una missió es pot veure en pantalla mitjançant una finestra petita que es mostra en tot moment. Aquest missatge [Figura 8.14] descriu breument el que ha de fer el jugador per avançar en el progrés de la missió. El jugador pot tenir més d'una missió iniciada al mateix temps, però pot triar-ne una que es mostri en aquesta finestra des del menú de pausa.

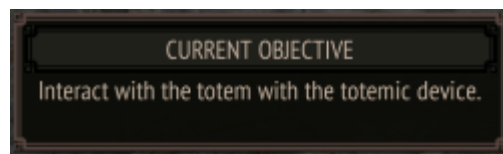


Figura 8.14. Finestra de missatge d'informació de la missió actual.

Molts elements del joc que s'exposaran més endavant tenen una estreta relació amb les missions dels jocs, ja que actualitzen l'estat d'aquestes i llegeixen l'estat per comprovar si el jugador compleix certes condicions.

8.3.2. Objectes

Repartides pel món, hi ha esferes lluminoses que representen objectes i que el personatge pot recollir quan hi entra en contacte. Un cop recollits, els objectes s'afegeixen a l'inventari del jugador i poden, o bé ser utilitzats per a algunes missions, o bé donar informació sobre l'assignatura objecte d'estudi. En aquest últim cas s'introdueixen noves entrades a l'*Archive*.

L'aspecte visual dels objectes [Figura 8.15] s'ha obtingut de l'*Asset Store* de *Unity*, del paquet *PowerUp particles* [39]. Són uns objectes amb un sistema de partícules que es pot crear a *Unity* i un *SphereCollider* amb l'opció *IsTrigger*.

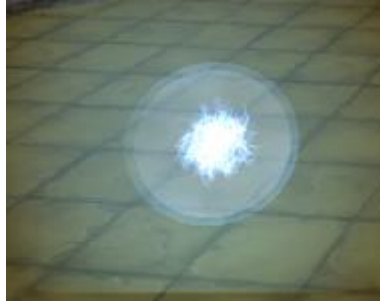


Figura 8.15. Esfera d'un objete en el joc.

S'ha creat un *Script* anomenat *Pickupable* amb diferents paràmetres, com el nom de l'objecte i la quantitat. A més, inclou el mètode *OnTriggerEnter*, el qual és cridat per *Unity* en el *frame* en el qual el *Collider* entra en contacte amb un altre. En aquest mètode, l'objecte s'afegeix a l'inventari del jugador, mitjançant la crida *PlayerController.AddItem* amb la informació del nom i la quantitat. Sempre que s'afegeix un objecte a l'inventari, surt un missatge a pantalla per notificar al jugador el que s'ha obtingut [Figura 8.16].



Figura 8.16. Finestra d'informació d'un objecte afegit a l'inventari.

Alguns objectes poden ser necessaris per completar algunes missions. En el cas que el jugador hagi començat la missió corresponent, en agafar l'objecte, també es comprova si ha aconseguit la quantitat necessària per a finalitzar la tasca i així poder retornar al *NPC* per tal d'obtenir un premi de reputació. Això també es produeix en el mètode *OnTriggerEnter* en què s'utilitzen els següents paràmetres per fer-ho:

- **questName:** Nom de la missió.
- **fromState:** Estat inicial necessari per avançar en la missió.
- **toState:** Estat següent si s'ha complert el requeriment.
- **requiredAmount:** Quantitat necessària d'objectes per completar la missió.

Finalment, un objecte també afegir nous continguts al menú *Archive* on es mostra la informació sobre l'assignatura, amb exemples interactius de cada tipus d'exercici. El funcionament és similar al de les missions en els objectes. Quan se n'afegeix un a l'inventari amb aquesta funcionalitat, es mostra un missatge a la pantalla [Figura 8.17].

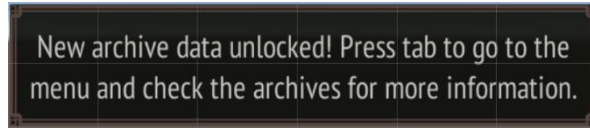


Figura 8.17. Missatge d'informació sobre nou contingut disponible al menú Archive.

8.3.3. Tòtems

El *tòtem* és l'element principal del joc, ja que és aquell amb què el jugador ha d'interactuar per tal de progressar i aconseguir els seus objectius. L'aspecte d'un *tòtem* és el que es pot observar a la Figura 8.18. En aquesta imatge, el *tòtem* té unes franges vermelles; això vol dir que es troba en l'estat desactivat. Si estigués activat, les franges serien blaves.



Figura 8.18. Aspecte dels tòtems.

Els *tòtems* poden tenir diverses funcions que cal diferenciar. Per una banda, pot tractar-se d'un *tòtem* que, en canviar el seu estat d'activació, fa moure un objecte de l'entorn, com per exemple una roca que bloqueja l'accés a un lloc. Per altra banda, hi ha *tòtems* que, en ser activats, fan aparèixer un objecte d'utilitat per a alguna missió del jugador. També poden canviar l'estat d'una missió i, en alguns casos, atorgar *reputació*. L'esquema de la Figura 8.19 mostra les *classes* que s'han creat per a afegir totes les funcionalitats necessàries als *tòtems*.

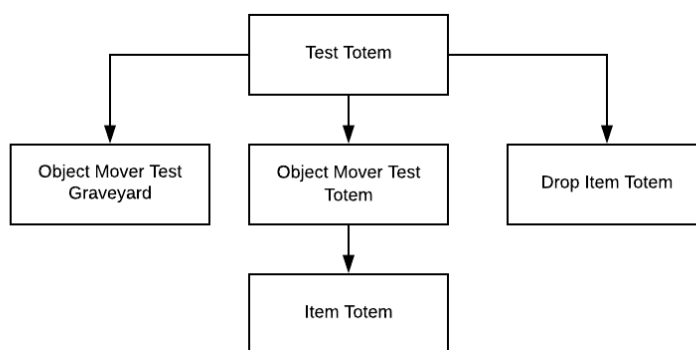


Figura 8.19. Esquema de les classes de tòtem que s'han creat.

El component *TestTotem* comprèn la funcionalitat bàsica dels tòtems. A diferència de la resta, aquest no fa res d'especial en activar-se o desactivar-se, excepte que canvia el color de les franges. Quan el personatge principal s'apropa al tòtem i entra en contacte amb el seu *Collider* (amb l'opció *Trigger*), es crida el mètode *OnTriggerEnter*, de forma similar a quan s'entra en contacte amb els objectes. En aquest cas, apareix una finestra a la part inferior de la pantalla que diu al jugador que pot prémer el botó de la lletra *E* per interactuar amb el tòtem [Figura 8.20]. Quan el jugador se'n va, es crida al mètode *OnTriggerExit* que desactiva aquesta finestra.



Figura 8.20. Exemple d'un jugador que pot interactuar amb un tòtem.

Com s'ha explicat anteriorment, per canviar l'estat d'activació d'un *tòtem* s'han de respondre correctament tots els tests interactius que apareixen en el moment que s'interactua amb ell. Aquests tests es desen a una variable *List<Test> testWindows* en ordre d'aparició. Així doncs, quan el jugador interactua amb el *tòtem*, apareix el primer test. Si la resposta és correcta, apareixerà el següent test, i així successivament fins que no n'hi hagi més o la resposta sigui errònia. Si totes les respostes són correctes, llavors canviarà l'estat d'activació del *tòtem*. És a dir, si les franges eren inicialment vermelles, després seran blaves, i viceversa. Quan s'activa i desactiva un *tòtem*, també apareix un missatge a la part central superior de la pantalla, la qual indica que s'ha respost correctament i que l'estat d'activació ha canviat.

A continuació, s'explica breument el funcionament de la resta de *tòtems*, que tenen una funcionalitat similar als *Test Totems*, però adaptada a diferents casos:

- **Object Mover Test Totem:** Té associat un objecte de l'entorn amb la variable *movableObject* i la posició final de l'objecte a la variable *target*. En activar-se, el *movableObject* es mourà a la posició *target*. Si es desactiva, torna a la posició inicial.
- **Object Mover Test Graveyard:** S'ha dissenyat per al nivell del cementiri, on hi ha il·luminacions diferents en activar els *tòtems*.
- **Item Totem:** Requereix que el jugador tingui una quantitat concreta d'un objecte a l'inventari per activar-lo. Si no la té, la finestra inferior passa a dir "*Can't use this object yet.*" (No es pot utilitzar encara aquest objecte). Quant a la resta, es comporta com un *Object Mover Test Totem*, que és capaç de moure un objecte de l'entorn.
- **Drop Item Totem:** El primer cop que s'activen, fan aparèixer un objecte. Els següents cops es comporten com un *Test Totem* en el qual no fan res especial.

Totes aquestes classes sobreescriven alguns mètodes de *Test Totem*, com en el cas de l'*Activate()*, que es crida quan s'activa un *tòtem*, mitjançant l'opció de *C# override*. A part, qualsevol *tòtem* pot ser utilitzat com a *tòtem elemental*; llavors en ser activat, s'actualitza l'estat de la missió principal *The Elemental Totems*. Aquest últim tipus de *tòtem* és dels més importants del joc, i per tant té un aspecte únic [Figura 8.21].



Figura 8.21. Imatges dels quatre tòtems elementals.

8.3.4. NPC

Durant el transcurs del joc, el jugador es troba amb diversos *NPC* que el poden guiar en el seu camí o pot demanar ajuda per a algunes tasques mitjançant missions. La implementació dels *NPC* té alguns aspectes similars a la del personatge principal, com és ara l'elecció d'un model 3D o la creació d'una *state machine* amb animacions. Tot i així, aquesta és més senzilla, perquè no cal tenir presents tants factors. En total hi ha 8 *NPC* en el joc i cadascun d'ells requereix un disseny personalitzat en molts aspectes. En general, tots ells tenen una implementació molt similar i es poden explicar de manera conjunta. El primer *NPC* amb què es troba el jugador, està situat al primer nivell, la cova. El seu model 3D, obtingut de *Mixamo*, es pot veure a la Figura 8.22 juntament amb altres models que s'han implementat al joc.



Figura 8.22. Tres dels models que s'han fet servir per als *NPC*.

De manera similar a la interacció amb els *tòtems*, quan s'entra en contacte amb el *Collider* amb funció *Trigger* del *NPC*, s'executa el mètode *OnTriggerEnter()*. Aleshores, apareix la finestra inferior que indica al jugador que pot prémer la tecla de la lletra *E* per interactuar amb ell. Si se'n va, es crida el mètode *OnTriggerExit()* i desapareix aquesta finestra. A la Figura 8.23 es veu al jugador davant de l'*NPC* amb la possibilitat d'interactuar-hi. Al mètode *Update()* es comprova si el jugador està suficientment a prop de l'*NPC* i si ha premut la tecla de la lletra *E*. En aquest cas, s'inicia la conversa.



Figura 8.23. Imatge del jugador amb la possibilitat d'interactuar amb un *NPC*.

Tots els *NPC* tenen un *Animator* amb un *AnimatorController*, que inclou una màquina d'estats molt simple amb només dos estats [Figura 8.24]. L'única variable que controla les transicions entre estats és *isTalking*, la qual val *True* si està parlant i *False* en cas contrari. Els estats que hi ha són:

- **Idle:** El *NPC* no fa res.
- **Talking:** El *NPC* està parlant i mou el cos amb l'animació corresponent.

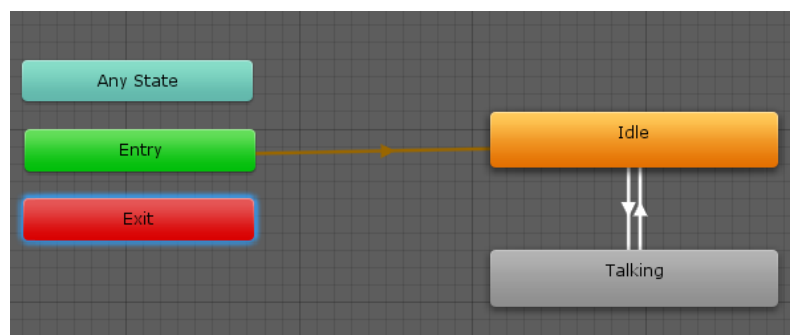


Figura 8.24. State machine dels *NPC*.

Les converses amb els *NPC* es mostren en format text en una finestra a la part inferior de la

pantalla [Figura 8.25]. Com pot haver-hi missatges llargs, el text està subdividit en missatges més curts i es pot avançar en la conversa polsant un botó.



Figura 8.25. Conversa del jugador amb un NPC.

Alguns NPC poden fer aparèixer un objecte necessari per continuar en la missió, de la mateixa manera que ho fan alguns tòtems. A més, tots els NPC tenen una missió associada i el que li diuen al jugador depèn de l'estat en què es trobi. Això es gestiona amb quatre variables de tipus *List<string>*, en les quals es desa el text corresponent a cada estat. Aquestes llistes són:

- **Text Messages:** És la seqüència de missatges que diu el NPC quan mai ha parlat amb el jugador. Sempre que es parla per primer cop amb un NPC, es comença una missió i mai es tornaran a dir els missatges de *Text Messages* si es torna a parlar amb ell.
- **Started Quest Messages:** Quan ja s'ha parlat amb el NPC però no s'ha completat la missió, són aquests els missatges que apareixen.
- **Completed Quest Messages:** En acabar la missió, el NPC dirà la seqüència de missatges que conté aquesta variable.
- **Final Messages:** Aquests missatges els dirà l'NPC quan ja li ha atorgat el premi de reputació, després d'haver completat la missió.

8.4. Tests interactius

La creació dels tests interactius es pot subdividir en dues categories. La primera és la creació de les finestres emergents que es mostren, amb la imatge del test i els botons, juntament amb altres elements de la UI. La segona categoria és el desenvolupament pel que fa al *Script* que

aleatoritza el contingut del test i comprova si el resultat de la resposta del jugador és correcte. En primer lloc, s'explica el procés de creació de la primera categoria.

8.4.1. Finestres de test

Les finestres de test tenen un aspecte similar a altres elements de la *UI*, com és ara, els tutorials i *tooltips* que s'han mostrat més amunt. A partir d'aquest estil, s'ha creat la finestra dels tests, amb un títol i un text informatiu sobre quina és la tasca a realitzar. Per als tests de circuits lògics, s'han creat les imatges dels circuits lògics i de les entrades i sortides del circuit mitjançant el programari *Inkscape* [34]. Després s'han incorporat a la finestra i s'ha posat especial èmfasi en fer que les imatges no es moguin ni distorsionin en ordinadors amb pantalles de mides diferents. Això s'ha fet mitjançant els *anchors* de *Unity*, que permeten ajustar la posició d'una imatge o element de *UI* en relació a la finestra, de manera que sempre es troben en la mateixa posició i la mida s'adapta a la pantalla. A la part inferior de la finestra, hi ha un botó que diu *Check* (comprova), que en ser clicat (o si es prem la tecla *E*) es comprova el resultat i s'avança al següent test, en cas d'haver-n'hi. Aquesta tasca la realitza el mètode *CheckResult*, que té assignat el botó i s'explicarà més endavant. A la Figura 8.26 es mostren dos exemples de tests interactius de circuits lògics: a l'esquerra, un test de portes lògiques bàsiques i a la dreta, un test d'un circuit més complicat.



Figura 8.26. Exemple de tests interactius.

Com que hi ha exercicis en els quals el jugador ha de fer clic sobre els elements de l'esquerra (entrades) i d'altres en què ho ha de fer sobre els de la dreta (sortides), pot dur a confusió sobre quins nivells demanen quina tasca. Per això, els elements que s'han de canviar parpellegen, mitjançant un canvi en el temps del seu factor de transparència *alpha*, de tal manera que es fan transparents i opacs contínuament. A més, el cursor es ressalta quan es passa per sobre d'aquests elements.

En relació als tests de les taules de la veritat [Figura 8.27], a un costat hi ha una taula de quatre o vuit files, depenent de l'exercici, a l'esquerra de la finestra. Cada element de la taula té un *Label* (línia de text) amb el nombre zero o u escrit dintre. La columna de la qual el jugador pot modificar els valors dels seus elements s'anomena *Y* i està composta de botons que tenen

la funció de modificar el text del seu camp. A l'altre costat de la finestra, hi ha un *Label* amb l'equació que ha de representar la taula.

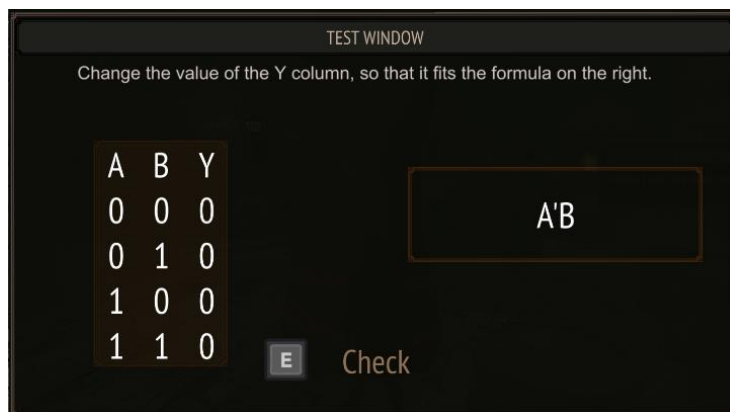


Figura 8.27. Exemple de test de taules de la veritat.

8.4.2. Funcionament dels tests

Tots els tests interactius deriven de la classe *Test* i tenen el mètode *CheckResult()*, el qual sempre és cridat quan es comprova el resultat. Segons el tipus de test, el mètode fa unes comprovacions diferents, ja que cada classe el sobreescrui. Per exemple, en els tests de portes lògiques simples, en les quals s'ajusta el valor de l'entrada per fer que la sortida tingui el valor lògic actiu, es comprova segons la porta lògica i el color de l'entrada (vermell o blau) si s'ha resolt correctament.

Els tests més complicats, en què es modifiquen els valors de les sortides d'un circuit, tenen associat un text que representa el valor que ha de tenir la solució. Per exemple, si en un circuit amb tres sortides la solució correcte és "010", les sortides han de valer (de dalt a baix, en ordre): Inactiu, actiu, inactiu. El mètode *CheckResult()* itera sobre les sortides i comprova, a partir del seu color, si el resultat és el correcte.

També cal destacar l'aleatorització de les entrades en els tests de circuits lògics. Els valors de les entrades no estan predefinits, sinó que es calculen quan es carrega la finestra en el mètode *Start()* i després defineixen el valor del resultat a partir d'això. En el cas de les taules de la veritat, s'aleatoritzen internament els valors de la columna *Y* corresponent al resultat correcte i després es crea la fórmula a partir, o bé dels *mintermes* o bé dels *maxtermes*, també de manera aleatòria.

8.5. Nivells del joc

A *Unity* existeix un tipus d'objecte especial anomenat *Terrain* (terreny en anglès), que serveix per crear la superfície d'un nivell. Es tracta d'una superfície en tres dimensions de gruix nul, que es pot modelar perquè pugui tenir diferents alçades, irregularitats i desnivells, gràcies a les eines que ofereix l'editor de *Unity*. A més, un *GameObject* amb un component *Collider* no pot travessar mai un terreny. Quan es crea un terreny, inicialment apareix com un pla de color blanc que, tal com s'ha dit, es pot modificar en alçada en certs punts per donar-li forma. A més, es pot afegir una *textura* per dibuixar el terreny, per així aconseguir l'aspecte del terra desitjat. Una altra funcionalitat és la d'afegir herba i arbres d'una manera similar a la d'afegir les *textures*. Un cop s'han afegit a l'editor, es pot seleccionar del menú l'arbre o l'herba a implementar i fer clic sobre la part del terreny on es vol afegir, a la *Scene View*. A part del terreny, per implementar altres elements com naturalesa i edificis als nivells, s'han utilitzat una gran varietat de paquets d'*assets* descarregats de la *Unity Asset Store*.

A més, cal mencionar com s'ha realitzat la transició d'un nivell a un altre. A cada nivell hi ha objectes amb el component de tipus *script*, anomenat *SceneTransition* i n'hi ha un per a cada possible *Scene* destí. En aquest *Script* s'inclouen totes les condicions que calen per poder accedir-hi, com requeriments de reputació o d'estat d'alguna missió. Quan toca fer la transició, es fa mitjançant una *Coroutine*, que funciona de manera similar a un *iterador* a *Python*. El mètode s'executa i quan arriba a la instrucció *yield*, es pausa l'execució i es retorna el control a *Unity* fins al següent *frame*, en el qual es continua per on la *Coroutine* ho havia deixat. Això es fa perquè s'ha introduït una funcionalitat, en la qual la pantalla es torna negra a poc a poc quan se surt d'una escena. Quan s'entra a la següent, la pantalla negra desapareix progressivament. A la *Coroutine* es crida el mètode *BeginFade*, que augmenta el paràmetre *alpha* de transparència d'una *textura* negra que ocupa tota la pantalla. Quan s'arriba al temps establert, el qual s'ha fet en el *yield* amb la crida '*yield return new WaitForSeconds(temps)*', la funció continua i es crida el mètode que canvia d'escena. A continuació, es mostra un exemple representatiu de cada nivell i alguns dels seus aspectes rellevants.

8.5.1. Cova

El primer nivell es pot dividir en dues parts: l'inici i les ruïnes de la civilització antiga. La part de l'inici és una cova formada bàsicament per roques gegants que formen un camí cap a les ruïnes. Aquí, s'ha aprofitat per ensenyar els controls del joc mitjançant *popups* que apareixen al llarg d'aquest camí. Els models de les roques es van obtenir del paquet *Rocks and Boulders 2* [40]. La Figura 8.28 mostra l'aspecte de l'entrada a la cova en el joc.



Figura 8.28. Exemple del nivell de la cova.

Pel que fa a les ruïnes, s'ha utilitzat el paquet *Cartoon Temple Building*, un conjunt de models i textures de terra, parets, decoracions i llums útils per crear masmorres i construccions d'ambientació antiga, com si es tractés d'un puzle [41]. A més a més, aquests models s'han aplicat per a la implementació d'alguns *tòtems* que s'encarreguen de moure els models en ser activats. Tot seguit, es mostra a la Figura 8.29 una part del nivell en la qual es pot veure l'aspecte de les ruïnes i un exemple d'aquests *tòtems* que poden moure els pilars, per permetre que el jugador pugui continuar el seu camí.

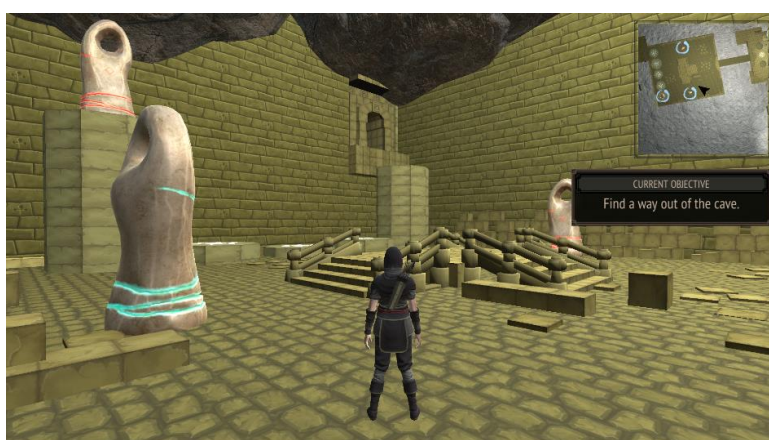


Figura 8.29. Exemple de les ruïnes al nivell de la cova.

8.5.2. Bosc

Per a la implementació del bosc [Figura 8.30], s'ha treballat sobre la base que ofereix l'asset *Rocky Hills Environment* [42], el qual ha estalviat molta feina i ha permès afegir més contingut útil al nivell. Aquest nivell predissenyat és molt extens i poc decorat; per això s'han afegit més elements decoratius i arbres de diferents tipus (obtinguts del *Nature Starter Kit* [43]), per tal

que el jugador també tingui un camí guiat per seguir cap al seu objectiu sense perdre's. També s'ha hagut de modificar el terreny del nivell i afegir *Colliders*, per evitar que el jugador pugui sortir dels límits del nivell.



Figura 8.30. Exemple del nivell del bosc.

8.5.3. Poble

El poble [Figura 8.31] és la zona central del joc. Està envoltat per quatre zones: el bosc, el cementiri, el port i el castell. A cada costat del poble hi ha una muralla i una porta d'accés a una zona i dins del poble hi ha cases i una plaça central. A més, hi ha decoració semblant a la del bosc, amb arbres i herba. Els edificis i la muralla s'han creat amb les eines que proveeix el paquet *Mega Fantasy Props Pack* [44].



Figura 8.31. Exemple del nivell del poble.

8.5.4. Cementiri

Aquest nivell [Figura 8.32] s'ha dissenyat a partir del paquet *Make Your Fantasy Game - Lite*, que conté una *Scene* amb un cementiri creat [45]. Amb tots els *assets* que conté aquest paquet, s'ha creat un nivell d'aproximadament el doble de grandària que el descarregat i s'han dut a terme moltes modificacions que han ofert la possibilitat de convertir-lo en un laberint amb els *tòtems* implementats. Cal destacar que aquest nivell inclou un altre model per als *tòtems* que encaixa més amb l'aspecte visual del cementiri.



Figura 8.32. Exemple del nivell del cementiri.

8.5.5. Port

En aquest nivell, també s'ha fet ús d'un *asset* gratuït de la *Asset Store*, anomenat *Old Sea Port* [46]. Inclou una *Scene* en la qual s'ha basat el nivell desenvolupat, tot i que han calgut molts canvis per adaptar-la al joc. S'han hagut d'afegir *Colliders* a totes les vores del nivell, per evitar que el jugador pugui caure a l'aigua o arribar a zones que no formen part de l'àrea de joc. Un punt interessant d'aquest nivell és l'aigua, que està constituïda per una superfície horitzontal i una *textura* que simula l'aspecte de l'aigua. La Figura 8.33 proporciona un exemple del nivell i de com es veu l'aigua dins del joc.



Figura 8.33. Exemple del nivell del port.

8.5.6. Castell

Per acabar, el castell és l'últim nivell del joc i engloba tots els tests que s'han fet fins al moment. Es comença el nivell a la sortida del castell [Figura 8.34] i s'hi entra a través d'un pont mitjançant l'activació del primer *tòtem*. El disseny del castell està basat en els mateixos assets que els del poble.



Figura 8.34. Exemple del nivell del castell.

8.6. Menú d'inici

El menú d'inici és la primera pantalla que veuen els jugadors quan inicien el joc. S'hi mostra el títol del joc i diverses opcions que pot fer l'usuari, per exemple: continuar la partida anterior, començar-ne una de nova o carregar una partida ja començada anteriorment. Així mateix, es pot triar el menú d'opcions per ajustar el volum de la música, dels efectes de so i per consultar informació del joc. La Figura 8.35 mostra l'aspecte del menú d'inici, el qual s'ha fet amb *GameObjects* amb components *Buttons* senzills de *Unity*.

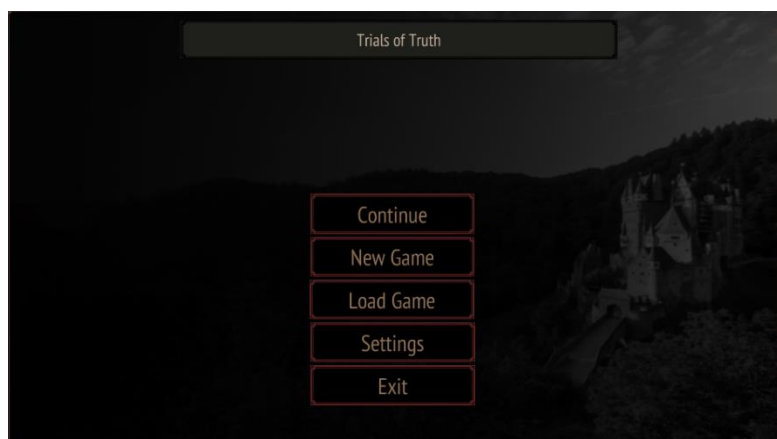


Figura 8.35. Pantalla del menú d'inici.

Els botons estan tots compresos dins d'un *GameObject*, de manera que quan aquest es desactiva mitjançant la crida *GameObject.SetActive(false)*, desapareixen tots ells. Així, quan per exemple es fa *clic* sobre el botó *Load Game*, es fan desaparèixer els botons i se n'activen d'altres. Totes les transicions entre diferents parts del menú d'inici s'han fet així.

No obstant això, els botons que han de passar a una altra *Scene* per carregar el nivell en el qual es troba el jugador, criden un mètode contingut a l'objecte *SceneTransition*, que funciona de la mateixa manera que per als nivells, tal com s'ha explicat més amunt.

8.7. Menú de Pausa

La implementació del menú de pausa s'ha realitzat utilitzant les eines de *Unity* per a la creació d'*UI*. El menú de pausa està compost per un total de cinc finestres (arxiu, missions, mapa, inventari i opcions), així com per una pestanya d'inici que apareix en prémer el botó de menú i permet a l'usuari seleccionar una finestra qualsevol.

La navegació entre finestres s'ha programat de la següent manera: Cada vegada que es produeix una transició entre finestres, es deshabilita la finestra actual i s'activa la següent. Quan s'entra en el menú, la pestanya d'inici s'activa. En tancar el menú, es desactiven totes les finestres.

Seguidament, es realitza una explicació detallada de cadascuna de les pestanyes que conformen el menú de pausa.

8.7.1. Finestra d'inici

Aquesta és la pestanya principal del menú de pausa i serveix de connexió amb la resta de finestres. Està dissenyada per un contenidor compost per botons.

El contenidor té un component anomenat *Horizontal Layout*, mitjançant el qual els botons ocupen sempre el màxim espai disponible, sense superposar-se, i es mantenen a la mateixa alçada.

Tal com s'ha dit, cada botó deshabilita la finestra actual i activa la següent. Així mateix, cada botó compta amb un component *Event Trigger* que permet executar unes instruccions de codi quan el punter entra, surt o clica en aquest objecte. Aquest component s'ha utilitzat per fer que la icona actual quedi ressaltada [Figura 8.36].

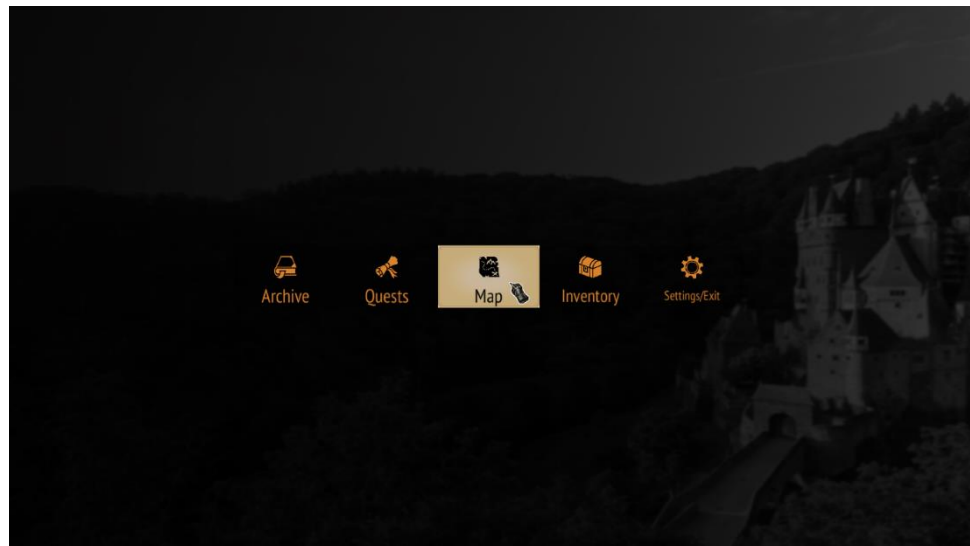


Figura 8.36. Pantalla del menú de pausa.

Per evitar que les icones segueixin ressaltades després de canviar de finestra, es comprova que retornin al seu color habitual tots els objectes, en el moment d'inicialitzar el menú de nou.

La resta de finestres reutilitzen aquest *Horizontal Layout* i el situen en la part superior de la pantalla. Així, no és necessari tornar a accedir a aquesta finestra fins que es torni a accedir al menú de pausa més endavant. A més, s'incorpora una petita barra a la dreta que inclou una visió ràpida del progrés del joc. Aquest disseny es pot veure a la Figura 8.37.

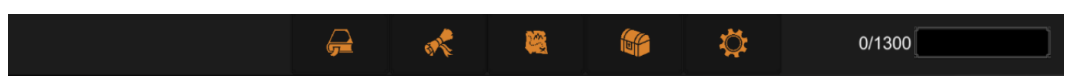


Figura 8.37. Esquema de menus horitzontal present a totes les finestres.

8.7.2. Arxiu.

La finestra arxiu [Figura 8.38] permet obtenir d'un cop d'ull els tutorials del joc i la informació docent disponible. Així mateix, la informació docent conté exemples interactius. Aquesta finestra està composta de dues parts: un índex a l'esquerra i una finestra d'informació a la dreta.

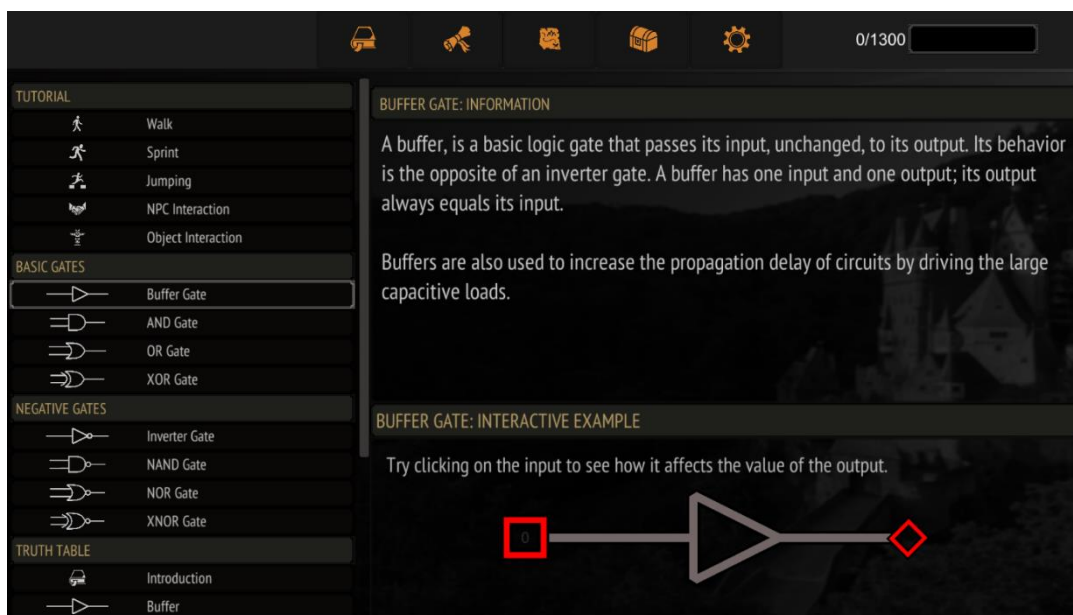


Figura 8.38. Pantalla de l'arxiu del menú de pausa.

L'índex a l'esquerra és un contenidor que afegeix contingut simultàniament amb els avenços del jugador. Per això compta amb dos components principals: *Vertical Layout* i *ScrollRect*.

De manera semblant al *HorizontalLayout*, el component *Vertical Layout* permet que cada nou apartat en l'índex se situï sota l'anterior. En aquest cas, cada component ocupa sempre el mateix espai fixat anteriorment. Per evitar el desbordament, el *Vertical Layout* s'introdueix a l'interior d'un *GameObject* amb un component *ScrollRect*. Gràcies a aquest element, si les categories superen l'espai assignat pel *ScrollRect*, és possible trobar els elements que no caben a la pantalla amb una barra lateral a la dreta.

Hi ha dos tipus d'elements dins de l'índex: Categories (*Tutorials*, *Portes bàsiques*, etc.) i els elements que els componen (*Caminar*, *Buffer*, etc.). Els *colliders* serveixen per agrupar la informació mitjançant un codi que permet amagar i desplegar tots els elements de la seva categoria. A la Figura 8.39 es pot veure com es comprimeixen i s'expandeixen totes les

pestanyes.

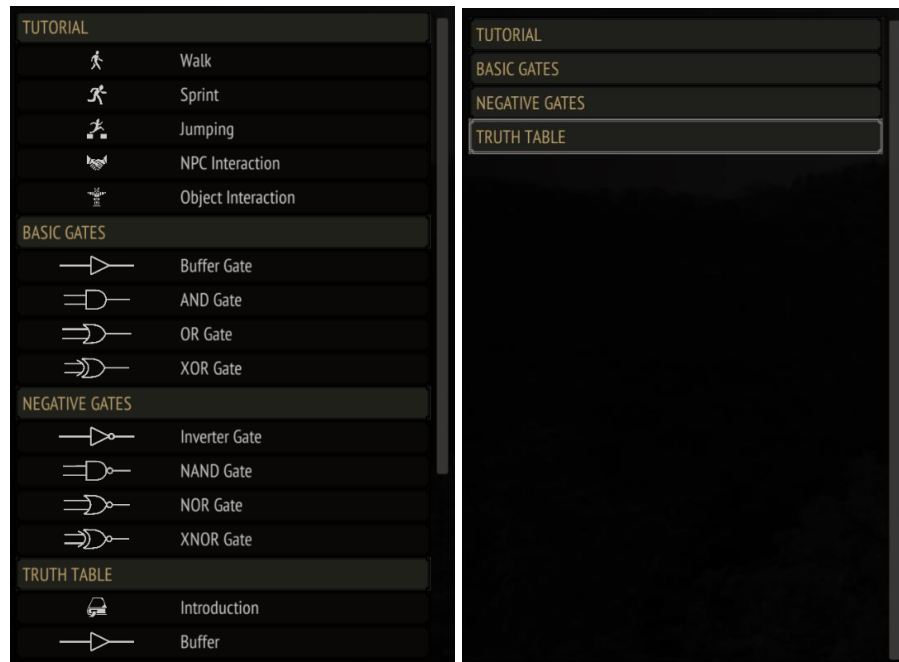


Figura 8.39. Exemple de l'índex expandit i comprimit.

Els elements de les categories contenen un *Script* que permet carregar *prefabs* i posar-los a la part de la dreta de la pantalla. El codi s'executa en fer *clic* en els elements de l'índex i ajusta la informació a la màxima grandària disponible. Quan es clica un nou element, l'objecte que estava instanciat anteriorment es destrueix.

Tant les categories com els elements que les componen, contenen un *Script* per afegir un marc al seu al voltant quan són premuts. D'aquesta manera, es proporciona més informació a l'usuari sobre les seves accions.

Evidentment, en funció del tipus d'element triat, la informació de la dreta varia. Els tutorials representen la mateixa finestra que les finestres que apareixen durant el joc. A part, les portes lògiques i les taules de la veritat contenen informació del seu funcionament i petits exercicis interactius. Aquests mini jocs es programen com un test que permet posar a prova el que s'ha après. A diferència dels puzles habituals, no existeix un botó per comprovar si el resultat és correcte, sinó que cada entrada de la porta lògica s'encarrega de modificar el seu color i el de la sortida, segons si li correspon estar activa o no. A la Figura 8.40, hi ha un exemple d'aquests mini jocs.

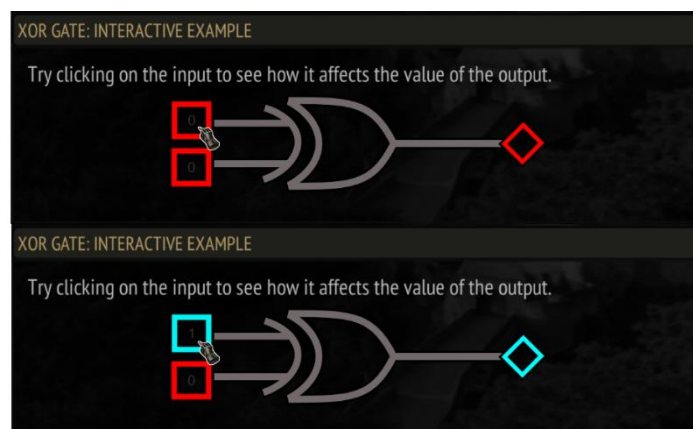


Figura 8.40. Tests interactius dins de l'arxiu.

Les taules de veritat contenen a més, la fórmula i la taula, tal com es pot veure a la Figura 8.41.

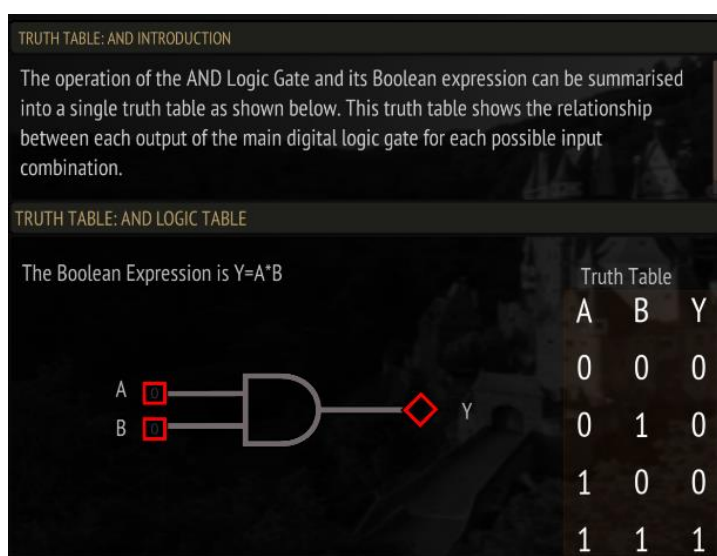


Figura 8.41. Taules de la veritat dins de l'arxiu.

El contingut del menú s'actualitza sempre que la finestra s'habilita. Per aconseguir-ho, aquesta té un codi que s'executa només quan s'activa (*OnEnable*). Així solament s'actualitza quan és necessari. Durant l'actualització, es llegeix el diccionari que conté tots els arxius que ha desbloquejat el jugador i s'afegeixen a la finestra aquells arxius que es troben disponibles.

8.7.3. Missions

Aquesta finestra [Figura 8.42] permet veure les missions començades i completades. Així mateix, també permet llegir-ne tota la informació del progrés i seleccionar la missió activa. De la mateixa manera que l'arxiu, la finestra està composta per dues parts principals: un índex a l'esquerra i una finestra d'informació a la dreta.

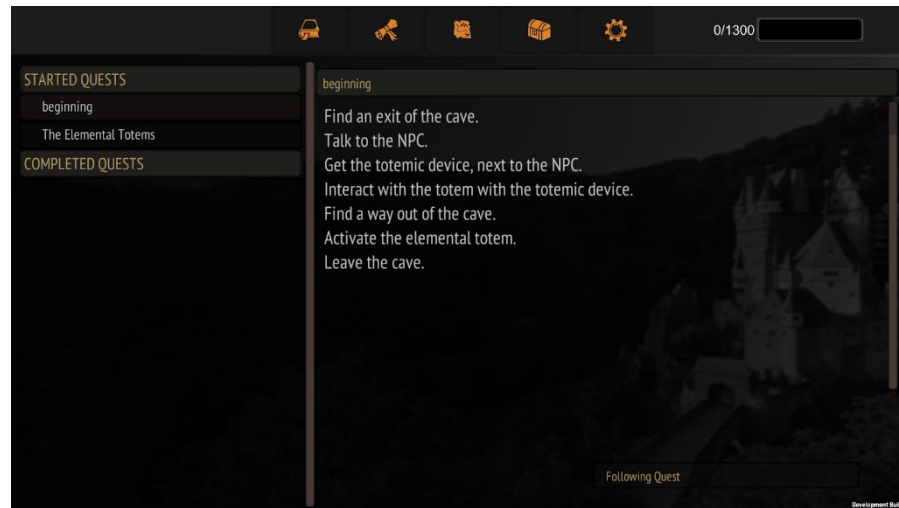


Figura 8.42. Finestra de missions dins el menú de pausa.

De manera similar que en l'arxiu, l'índex de l'esquerra està compost per un contenidor amb el component *ScrollRect* i *Vertical Layout*. Com s'ha vist anteriorment, gràcies a aquests components, és possible afegir nous elements de manera molt senzilla.

El contenidor està compost de dos tipus d'elements, com a l'arxiu: Categories i els elements que les componen. Quan es clica sobre el títol de la categoria, s'amaguen o es despleguen tots els seus continguts.

La finestra de la dreta conté la informació actualitzada de la missió i permet seguir la que es desitgi. Aquesta selecció permet visualitzar la missió en la *UI*. A la Figura 8.43 es pot veure el botó de seguir missions abans i després de segui-ne una.

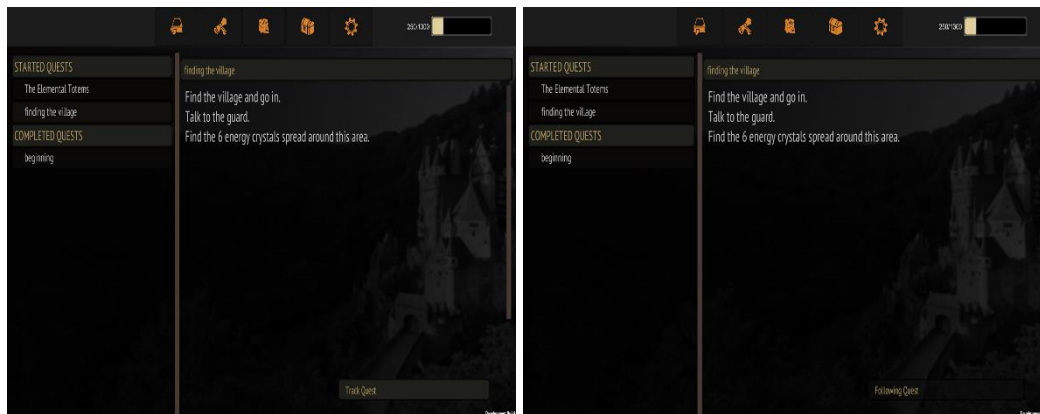


Figura 8.43. Botó de seguir missió dins de la finestra de missions.

Quan s'activa la finestra de missions, es llegeix un diccionari amb l'estat de les missions i es carreguen. Si una missió s'ha completat, passa de la categoria de missions començades a la de missions completades. A la Figura 8.44 es pot veure com la missió *beginning* és a la categoria de les que ja s'han completat.

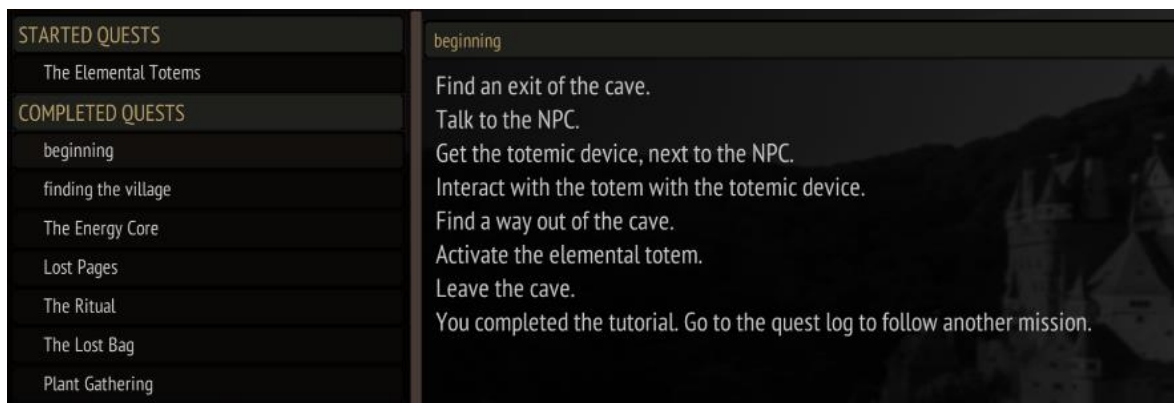


Figura 8.44. Exemple de missions a la secció de missions completades.

De manera diferent que a l'arxiu, a la finestra missions existeix un objecte base ocult que es copia cada cop que es comença una missió. Aquest element copiat rep un canvi de nom, es fa visible i s'afegeix a la categoria de missions iniciades.

El text de la finestra de la dreta s'actualitza automàticament quan se selecciona una missió. Només cal actualitzar el text que es mostra a partir de l'estat del jugador a la missió i es llegeix d'un diccionari el text que conté les descripcions de les missions.

8.7.4. Mapa

La finestra del mapa permet obtenir una visió aèria global del nivell actual. En aquesta imatge es representen les icones més interessants, així com la posició del jugador en temps real. A la Figura 8.45 es pot veure una imatge de la finestra del mapa al nivell de la cova.



Figura 8.45. Pantalla de mapa dins del menú de pausa.

Per aconseguir el disseny d'aquesta finestra, s'afegeix una càmera addicional al joc en perspectiva ortogonal que engloba tot l'escenari. Aquest mètode, molt similar al que s'ha emprat per al mini mapa, permet mantenir la informació del joc constantment actualitzada.

Les icones del mapa s'afegeixen a l'escenari fent servir un *quad*. Un *quad* és un objecte 3D de Unity que s'assembla a un pla i és útil per visualitzar una imatge en un entorn tridimensional. A aquests objectes, se'ls assigna un *Layout* de mini mapa i s'estableix que solament seran renderitzats per la càmera del mapa i del mini mapa.

La imatge de la càmera s'exporta com *textura* i se situa en un rectangle a la pantalla. A aquesta imatge se li adjunta un codi que permet fer zoom i desplaçar el mapa. Es pot veure un exemple del mapa amb zoom a la Figura 8.46.



Figura 8.46. Mapa del nivell amb zoom.

Finalment, per evitar un consum massa alt de recursos del joc, només s'activa la càmera quan la finestra del mapa està oberta.

8.7.5. Inventari

L'inventari [Figura 8.47] permet obtenir una visió de tots els objectes que s'han obtingut fins al moment, així com la seva quantitat. Està format principalment per un *Vertical Layout* que permet afegir més objectes fàcilment.



Figura 8.47. Pantalla de l'inventari dins del menú de pausa.

L'inventari actualitza automàticament el *vertical layout* en activar-se. El procés consisteix a

llegir un diccionari associat al jugador, compost pels objectes que ha obtingut. A la Figura 8.48 es pot veure la mateixa finestra del cas anterior, però just després de recollir un objecte addicional.

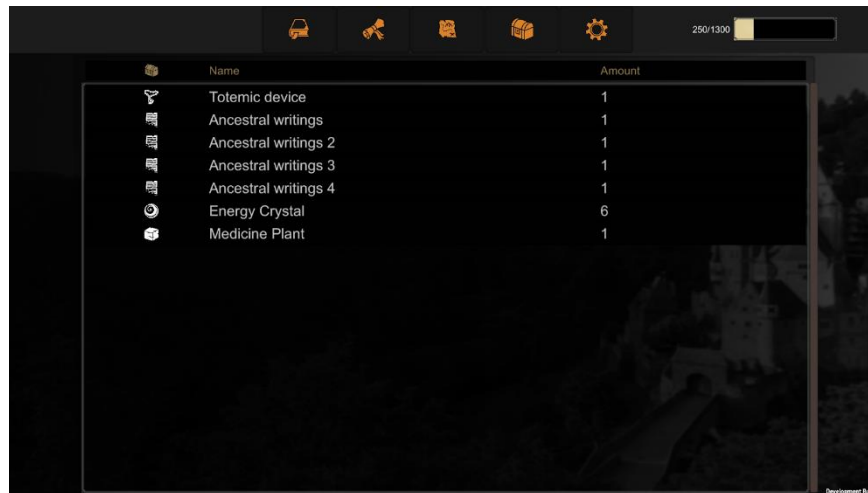


Figura 8.48. Finestra de l'inventari quan després de recollir un altre objecte.

8.7.6. Menú d'opcions

El menú d'opcions [Figura 8.49] permet abandonar la partida, canviar el volum de la música i conèixer més coses sobre els desenvolupadors. Està format per un *VerticalLayout*, compost de botons i *sliders*.

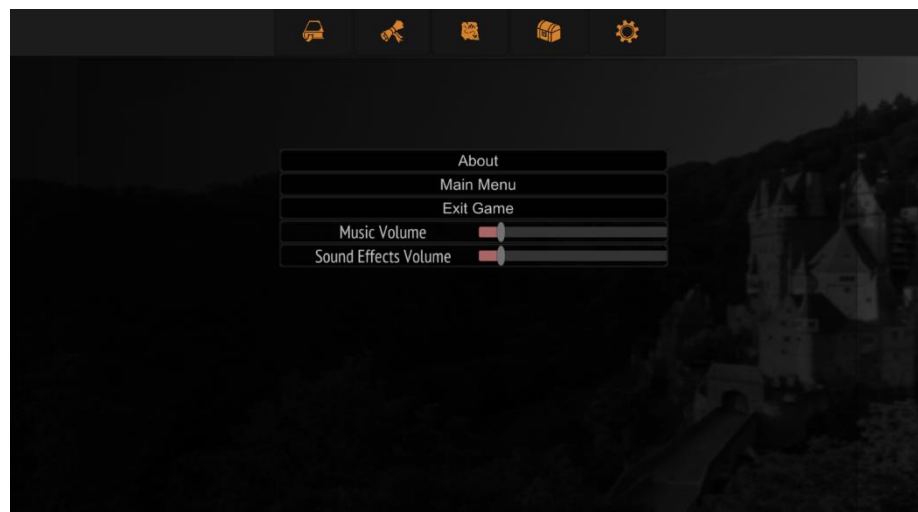


Figura 8.49. Finestra d'opcions dins del menú de pausa.

El botó *about* obre una finestra emergent amb informació dels desenvolupadors del projecte. El funcionament per mostrar i ocultar la finestra consisteix en l'habilitació i desactivació de

l'objecte mitjançant l'ús de botons. Si tot el menú es tanca, la finestra es desactiva automàticament. A la Figura 8.50. Secció sobre els creadors del joc dins de les opcions. es pot veure una imatge de la finestra emergent.

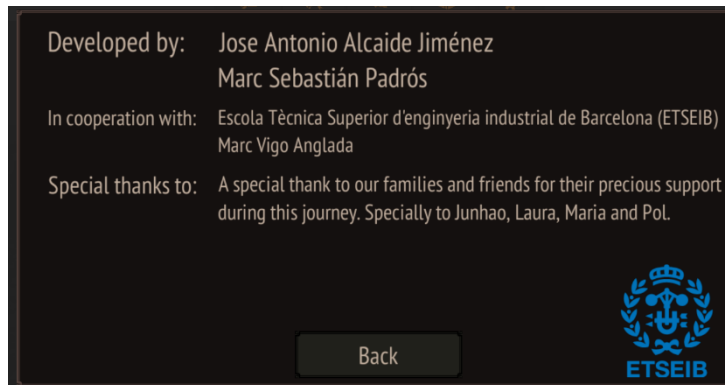


Figura 8.50. Secció sobre els creadors del joc dins de les opcions.

Els botons *main menu* i *exit game* permeten anar cap al menú principal i sortir del joc respectivament. El seu funcionament és idèntic al de la finestra *about*. A la Figura 8.51. Finestres emergents per sortir del joc. es mostren ambdues finestres emergents.



Figura 8.51. Finestres emergents per sortir del joc.

El volum de la música i el volum dels efectes de so es controlen mitjançant la utilització de *sliders* [Figura 8.52]. Cada *slider* modifica el valor d'una variable estàtica de *GameInfo*, la qual es consulta a l'hora de reproduir la música i els efectes de so.

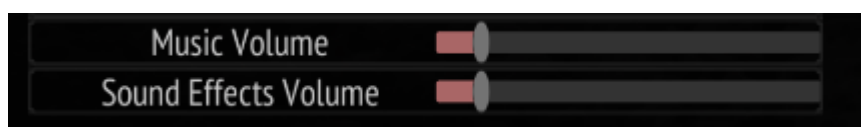


Figura 8.52. Sliders de control de volum en la secció d'opcions.

8.8. Música i Efectes de So

Unes de les parts més importants d'un videojoc són la música i els efectes de so, ja que tenen la capacitat d'estimular l'interès del jugador en el joc i fer-lo més agradable. Per això s'ha fet especial èmfasi en què sigui variada, perquè no sigui repetitiva. Amb tot, la temàtica de la música segueix una mateixa tendència en el gènere i tipus de música. Tots els elements de so s'han obtingut de la *Unity Asset Store* i de *Opengameart.org* i funcionen a *Unity* mitjançant el component *AudioSource*. Aquest reproduïx el so indicat quan es crida el mètode *Play* del *GameObject* que el conté. Cada objecte amb el component *AudioSource* pot ser regulat en diversos aspectes, com el volum i la forma en què es reproduïxen.

Cada *Scene*, és a dir, cada nivell i el menú d'inici, tenen música de fons que es repeteix en bucle. Pel que fa als efectes de so, cada efecte depèn de l'objecte al qual està associat. Per exemple, un *tòtem* fa un so en ser activat per indicar que la resposta ha estat correcta. Quan un objecte que mou el *tòtem* es mou per alguna superfície, també se sent un so determinat.

L'efecte de so amb més interès és el que es genera quan el personatge camina. A cada pas, es reproduïx un so que representa el soroll de les sabates amb el terra. Segons el tipus de superfície, el so és diferent; es distingeix entre herba, formigó, fusta, sorra i metall. Per implementar aquest sistema, s'ha utilitzat l'asset *Free Footsteps System* [47], el qual fa servir un objecte anomenat *footsteps_database* on es relacionen tots els noms de les *textures* del joc amb el seu corresponent tipus de terra. A més, per cada tipus de terra hi ha una llista de sons per alternar entre ells quan el personatge camina. A cada peu del personatge s'ubica un *GameObject*. El *Script* del *Footsteps System* reproduïx un so aleatori de la llista associada al tipus de terra sempre que hi entra en contacte, de manera que té una semblança propera a la realitat.

8.9. Controls del Joc

Per ajustar els controls del joc, *Unity* permet assignar mètodes d'entrada a tipus d'accions, com per exemple l'acció *Saltar*, a la qual se li pot associar el botó d'espai. Després, a l'hora de validar si s'ha premut una tecla, s'utilitza la crida *Input.GetKeyDown* i es passa el nom del botó com a variable, el qual és de la classe *KeyCode*. Per exemple, la tecla 'E' seria *KeyCode.E*.

En resum, els controls del joc són els següents:

- **Córrer:** Tecles *W*, *A*, *S* i *D*.
- **Girar la càmera:** Moviment del ratolí.
- **Caminar cap endavant:** *Shift*.

- **Esprintar:** Còrrer prement *Shift*.
- **Interactuar:** *E*.
- **Saltar:** Barra d'espai.

8.10. Emmagatzematge de dades

En tractar-se d'un joc amb una duració d'aproximadament dues a tres hores, convé que els jugadors puguin desar el seu progrés per reprendre el joc en qualsevol altre moment. El sistema que s'ha triat per gestionar la informació que ha de llegir el joc en el moment de carregar partida, és mitjançant fitxers de format *XML* (*eXtensible Markup Language*). Un fitxer *XML* desa dades com qualsevol altre fitxer de text, però d'una manera molt ordenada, en la qual la informació es pot categoritzar.

Quan es compila un projecte a *Unity*, totes les dades es converteixen en fitxers d'un altre format i no són editables, excepte en el cas dels fitxers provinents d'una carpeta anomenada *Streaming Assets*. La carpeta i tots els seus continguts són copiats quan s'ha compilat el joc, de manera que és molt útil tenir fitxers de dades de la partida del jugador dins d'aquesta carpeta, ja que es pretén accedir i modificar aquests fitxers.

S'ha creat un fitxer de nom *sample.xml* que conté l'estructura en què s'organitzen les dades que necessita el programa per carregar la informació del jugador. A més, tota la informació sobre les missions del joc és al fitxer *quests.xml*, en el qual estan definits els estats inicials i finals de cada missió, i tots els textos de descripció referents a cada estat.

El joc desa i carrega les dades a una carpeta de l'usuari que depèn de cada ordinador. Normalment, el camí és "*C:/Users/<Nom_usuari>/AppData/LocalLow/DefaultCompany/Trials Of Truth*", on *<Nom_usuari>* és el nom de l'usuari al compte de *Windows*. El primer cop que s'inicia el joc, es copien els fitxers des de la carpeta *Streaming Assets* al fitxer de l'usuari, al qual s'accedeix a *Unity* mitjançant la crida *Application.persistentDataPath*. Quan un jugador es crea un usuari, es copia el fitxer *sample.xml* amb el nom d'usuari que s'hagi triat a la carpeta destí.

A cada *Scene* de *Unity* s'ha creat un objecte de nom *SaveLoad* que s'encarrega de gestionar l'emmagatzematge de dades, la càrrega de partides i la càrrega de cada escena quan es fa una transició des d'una a l'altra. Per una banda, el mètode *LoadQuests* llegeix del fitxer *quests.xml* tota la informació sobre les missions i la posa a objectes de classe *Quest*. La variable estàtica *GameInfo.quests* conté tots aquests objectes de manera que són accessibles per qualsevol altre objecte. Per altra banda, *LoadPlayer* llegeix tota la informació del jugador:

l'estat en cada missió, els objectes al seu inventari, la informació disponible a l'*Archive*, etc. Tota aquesta informació es desa a *GameInfo* o *PlayerController* en variables estàtiques, per a ser llegibles més endavant. La posició del jugador no es desa enlloc, però en el moment que es llegeix, es col·loca el jugador en aquelles coordenades, fent que el *transform.position* de la instància del jugador tingui els valors corresponents.

Per desar partida, es duu a terme el mateix procés que per carregar, però a la inversa. La idea és incorporar les dades rellevants de *GameInfo* i *PlayerController* al fitxer *xml* del jugador, substituint les antigues. El mètode *SavePlayer* és el que realitza aquesta tasca, però hi ha un aspecte important que cal tenir en compte, i és que no sempre s'ha de poder desar partida. Si un jugador es trobés en una situació a la qual hagués arribat a causa d'algun error del joc i guardés partida en aquell moment, estaria atrapat per sempre, ja que en carregar tornaria a aparèixer allà mateix i no se'n podria sortir. Per això s'utilitzen punts de desat, és a dir, zones concretes on es permet que el jugador desi l'estat de la seva partida. Dins d'aquesta posició, es pot prémer un botó que dóna la instrucció de cridar el mètode *SavePlayer*. L'aspecte gràfic dels punts de desat, que s'han creat amb ajuda del paquet *UET Particles Kit* de la *Unity Asset Store*, es pot veure a la Figura 8.53, al costat del jugador. A més, se l'hi ha afegit un component de partícules grogues flotant *Particle System* que li atorguen un disseny més interessant. [48]



Figura 8.53. Punt de desat i procés de desar partida.

Quan el jugador entra en contacte amb el *Collider* d'un punt de desat, apareix un missatge que indica que es pot guardar la partida si es polsa el botó de la lletra *E*. Si es fa, el joc desa la partida i apareix un missatge dient que s'ha desat partida satisfactoriament, tal com es mostra a la figura anterior. Hi ha punts de desat repartits pel món, perquè el jugador pugui guardar el seu progrés en diverses situacions.

9. Validació

9.1. Intenció de la validació

En qualsevol desenvolupament de producte, és necessari realitzar una exhaustiva verificació de les seves qualitats abans de finalitzar l'etapa de desenvolupament. Això és especialment important en el camp del programari en què les proves de validació permeten descobrir defectes del programa. Motivats per aquesta necessitat, al termini del projecte hem dut a terme una prova de verificació.

D'una banda, l'objectiu és descobrir tots aquells errors tècnics, com per exemple errors de codi o de rendiment. D'altra banda, també es cerquen possibles errors gràfics (parpelleig de textures, errors en la il·luminació, etc.) o de disseny (zones inaccessibles o de les quals no es pot sortir, etc.). La resolució d'aquests errors és un aspecte fonamental i s'estableix com a prioritària.

Endemés, es pretén descobrir si s'ha complert l'objectiu del joc plantejat a l'inici del projecte; això és, aconseguir un joc complet, divertit i educatiu. L'anàlisi d'aquesta informació té un pes molt important de cara a la implementació de futures millores.

9.2. Qüestionari

Les proves de validació es realitzen en un grup reduït de persones de diversos perfils. Als usuaris se'ls proporciona la informació per descarregar-se el joc i han de respondre un qüestionari com a mínim després d'aconseguir completar el primer nivell.

Els perfils dels usuaris són diferents i no es limita a persones amb coneixements d'electrònica. Així mateix, tampoc no es limita a persones amb experiència amb jocs d'ordinador per tal de simular el màxim possible una possible implementació real. Per no contaminar la informació, als usuaris no se'ls guia durant el joc, tot i que se'ls facilita suport si així ho desitgen.

La majoria de les preguntes que conformen l'enquesta apliquen *l'escala de Likert*. Aquesta escala s'utilitza per conèixer el grau de conformitat d'una persona amb una determinada oració que afirma o nega alguna cosa concreta. Les respostes possibles segons *l'escala de Likert* segueixen una estructura lineal amb cinc opcions, que varien des de *totalment en desacord* fins a *totalment d'acord*, amb respostes intermèdies.

Les preguntes es van organitzar en quatre categories: *Introducció, aprenentatge, disseny i mecàniques i conclusions*. A la Taula 9.1, es presenten les preguntes del qüestionari que s'ha creat amb *Google Forms*. [49]

Núm.	Pregunta	Resposta
Introducció		
1	Quins són els teus estudis?	Desplegable
2	Tenies coneixements d'electrònica abans de provar el joc?	Cert / Fals
3	Sóc jugador habitual de videojocs d'ordinador.	Escala de Likert
Aprenentatge		
4	He adquirit coneixements de la matèria (electrònica).	Escala de Likert
5	La matèria explicada en el joc m'ha semblat interessant.	Escala de Likert
6	La matèria està correctament explicada.	Escala de Likert
7	El nivell de dificultat de les preguntes de test augmenta a un nivell correcte	Escala de Likert
Disseny i mecàniques de joc		
8	El joc és visualment atractiu.	Escala de Likert
9	M'ha agradat el disseny de l'entorn i els nivells.	Escala de Likert
10	La història del joc m'ha semblat interessant.	Escala de Likert

11	Les missions del joc m'han semblat interessants.	Escala de Likert
12	Sempre he tingut clars els objectius dins del joc	Escala de Likert
13	La música i el so afegeixen interès al joc.	Escala de Likert
14	El control del personatge m'ha semblat fàcil.	Escala de Likert
15	La interacció amb els puzles m'han semblat interessants.	Escala de Likert
16	La navegació pels menús és fàcil i intuïtiva.	Escala de Likert
17	He trobat fàcilment la informació referent a la matèria a l'arxiu (secció archive).	Escala de Likert
Conclusions (Resum)		
18	M'ha semblat un joc divertit.	Escala de Likert
19	M'ha semblat un joc útil.	Escala de Likert
20	Incorporaria jocs d'aprenentatge com aquest per a aprendre conceptes universitaris.	Escala de Likert
21	Recomanaria aquest joc a un estudiant d'electrònica.	Escala de Likert
22	Què és el que més t'ha agradat?	Text (Opcional)
23	Què és el que menys t'ha agradat?	Text (Opcional)
24	Suggeriments addicionals	Text (Opcional)

Taula 9.1 Preguntes del qüestionari

9.3. Resultats

Els qüestionaris s'han dut a terme amb un grup de 14 persones (12 universitaris, un estudiant de batxillerat i un estudiant de secundària). D'aquest grup, només un 57,1% de persones tenien coneixements d'electrònica [Figura 9.1]. Per altra banda, l'experiència amb els jocs ordinador que tenen és molt diversa [Figura 9.2].

Tenies coneixements d'electrònica abans de provar el joc?

14 respuestas

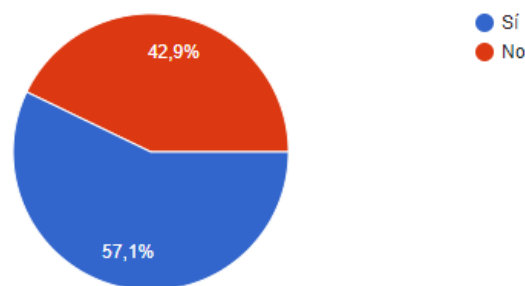


Figura 9.1. Gràfic sobre els coneixements d'electrònica dels enquestats.

Sóc jugador habitual de videojocs d'ordinador.

14 respuestas

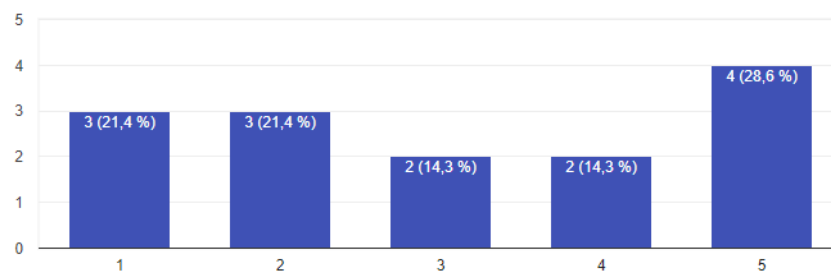


Figura 9.2. Experiència amb els jugadors d'ordinador dels enquestats.

Per avaluar els resultats dels tests, s'assigna a cada resposta una numeració de l'1 al 5 (1 significa *Totalment en desacord* i 5 significa *Totalment d'acord*). A la Taula 9.2 es mostren els resultats dels tests:

Núm.	Pregunta	Resposta
Aprenentatge		
4	He adquirit coneixements de la matèria (electrònica).	3,71
5	La matèria explicada en el joc m'ha semblat interessant.	4,21
6	La matèria està correctament explicada.	4.57
7	El nivell de dificultat de les preguntes de test augmenta a un nivell correcte	4.5
-	Resum Aprenentatge	4.25
Disseny i mecàniques de joc		
8	El joc és visualment atractiu.	4.43
9	M'ha agradat el disseny de l'entorn i els nivells.	4.57
10	La història del joc m'ha semblat interessant.	4.43
11	Les missions del joc m'han semblat interessants.	4.07
12	Sempre he tingut clars els objectius dins del joc	4.29
13	La música i el so afegeixen interès al joc.	4.79
14	El control del personatge m'ha semblat fàcil.	4.00
15	La interacció amb els puzles m'han semblat interessants.	4.50
16	La navegació pels menús és fàcil i intuïtiva.	4.57
17	He trobat fàcilment la informació referent a la matèria a l'arxiu (secció archive).	4.79
-	Resum Disseny i mecàniques de joc	4.38
Conclusions (Resum)		
18	M'ha semblat un joc divertit.	4.07
19	M'ha semblat un joc útil.	4.43
20	Incorporaria jocs d'aprenentatge com aquest per a aprendre conceptes universitaris.	4.00
21	Recomanaria aquest joc a un estudiant d'electrònica.	3.79
22	Què és el que més t'ha agradat?	-
23	Què és el que menys t'ha agradat?	-
24	Suggeriments addicionals	-
-	Resum Conclusions	4.07
-	Resum General	4.24

Taula 9.2 Resultats dels tests

Si s'analitza de manera global, el joc té una mitjana superior a quatre sobre cinc, la qual cosa implica un resultat satisfactori. De cara a l'aprenentatge, es pot veure que la puntuació sobre si s'han adquirit coneixements d'electrònica és inferior a quatre, tot i que la tendència encara és positiva [Figura 9.3]. Aquest fet podria ser en primer lloc per la limitació de temari, ja que no abraça molts conceptes de l'assignatura d'electrònica. No obstant això, la majoria coincideixen en el fet que la matèria està ben explicada [Figura 9.4], és interessant i augmenta de dificultat en una progressió adequada.

He adquirit coneixements de la matèria (electrònica).

14 respuestas

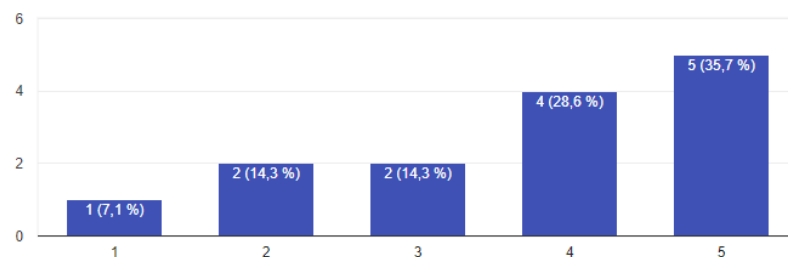


Figura 9.3. Respostes sobre si s'han adquirit coneixements d'electrònica amb el joc.

La matèria està correctament explicada.

14 respuestas

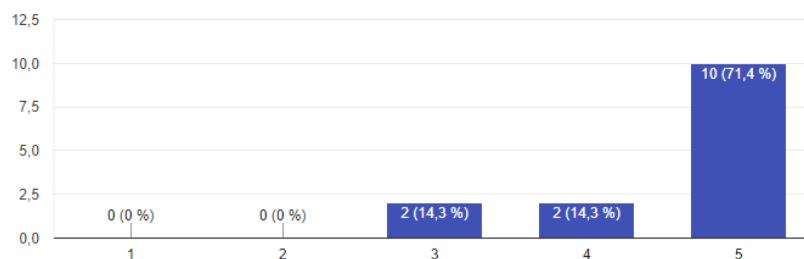


Figura 9.4. Respostes sobre si la matèria esta correctament explicada.

En relació a les mecàniques de joc, cal mencionar que el control del personatge és allò que més ha costat als enquestats i que les missions del joc tenen marge de millora. Pel que fa a la resta de mecàniques i disseny del joc, l'aspecte visual i la música han tingut una bona recepció, així com el disseny dels nivells. Generalment, s'indicaven correctament els objectius del personatge en tot moment i la informació referent a l'assignatura d'electrònica era fàcil de trobar.

Finalment, la majoria de jugadors ha trobat el joc divertit [Figura 9.5] i útil [Figura 9.6]. Tanmateix, no tots recomanarien el joc o d'altres jocs seriosos per a aprendre conceptes universitaris, segurament a causa d'una falta de maduresa del joc, tant en temari com en mecàniques.

M'ha semblat un joc divertit.

14 respuestas

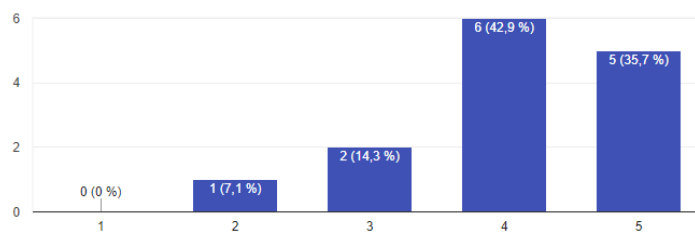


Figura 9.5. Opinió dels enquestats sobre si el joc és divertit per a ells.

M'ha semblat un joc útil.

14 respuestas

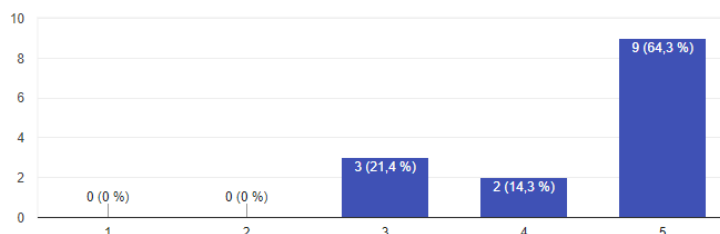


Figura 9.6. Resposta dels enquestats sobre la utilitat del joc per a l'educació.

Quant a les respostes de suggeriments, el que més ha agradat és la temàtica, l'aprenentatge d'electrònica i el disseny gràfic del joc. Tot seguit, mostrem alguns comentaris destacats sobre allò que més ha agradat als enquestats:

- El fet que a mesura que vas avançant en el joc, ets capaç d'entendre millor com funcionen les portes lògiques.
- Els puzles que requerien pensar quins *tòtems* s'han d'activar o desactivar i en quin ordre cal fer-ho per progressar (en comptes d'activar-los tots indiscriminadament per avançar).
- Els reptes constants dels *tòtems* al joc m'han agradat molt, ja que ajuda a interioritzar els tipus de circuits i recordar-los al moment. A diferència de l'aprenentatge tradicional, en el joc la recompensa és immediata.
- El fet d'aprendre mentre es juga està molt ben integrat en el joc. Realment es pot aprendre sense adonar-se'n ni avorrir-se, també gràcies a una història interessant i un bon fil conductor.

- Aprendre electrònica sense adonar-me'n, com si fossin enigmes. Que el personatge fos una noia. Disseny molt bonic.
- M'ha agradat la història que tenia bona pinta, es feia amena i els diàlegs amb els personatges. M'he quedat amb ganes de jugar, això és bo.
- Les missions, el disseny gràfic, i l'àmbit educatiu del joc.

Per altra banda, la majoria d'enquestats coincideixen que el que menys els ha agradat ha sigut el control del personatge. També destaquen comentaris envers el mapa i els marcadors que podien ser més clars. Vegem algunes opinions negatives sobre el joc:

- Des del menú no es poden veure els mapes de zones on no hi ets. No hi ha marcadors de missions. No hi ha diferència entre tòtems actius i inactius en el mapa. Només un canvi de color de la icona en el mini mapa seria suficient.
- Control del personatge, sobretot a llocs on s'ha de saltar.
- A vegades la informació venia molt de cop quan potser encara no t'havies habituat a algun dels conceptes.
- Que estigués exclusivament en anglès. Al principi no m'ha semblat fàcil el control del personatge.
- Cap al final del joc es fa una mica pesat resoldre els tests i es perd la concentració; això fa que es cometin més errades.

En els suggeriments addicionals els enquestats han donat la seva opinió sobre alguns dels problemes del joc i han proposat possibles solucions. Vegem alguns comentaris:

- En els tests de les taules de la veritat, si surt una pregunta molt complicada i no es vol pensar, es pot sortir del test i tornar-hi a entrar fins que surt una pregunta més senzilla.
- Els *tòtems* del cementiri haurien de ser més visibles o diferenciables amb l'entorn.
- Cap al final del joc, es podrien introduir els *flip-flops* o *comptadors* que s'haurien de muntar a partir de les portes bàsiques per resoldre algun problema en el poble. Així canviaria el mètode d'aprenentatge i es faria més amè.
- Més idiomes.
- Poder sortir del menú de pausa amb el botó *Escape*.

10. Reptes i dificultats del projecte

El desenvolupament d'aquest projecte ha tingut sens dubte etapes en les quals hi ha hagut moltes dificultats, tant en el moment del disseny com en el de la implementació. Aquest apartat destaca els majors reptes que hi ha hagut en tot el desenvolupament.

En primer lloc, la tria del tema d'aprenentatge va suposar un repte que es va trigar molt temps a decidir, ja que és complicat ensenyar una assignatura en un joc interactiu i que sigui interessant per al jugador. A més, es volia posar èmfasi en el fet que no fes falta fer servir cap altre material (com paper i bolígraf) per poder calcular els resultats dels exercicis. Inicialment, s'havia pensat a triar l'electrotècnia com a tema d'aprenentatge i fer tests interactius amb circuits elèctrics, però la complexitat dels exercicis en nivells de dificultat lleugerament superiors al més bàsic van propiciar el canvi al tema que finalment es va triar, les portes lògiques d'electrònica.

En segon lloc, tot el projecte s'ha programat en un llenguatge inicialment desconegut pels integrants del grup, *C#*. Aprendre un llenguatge nou mentre es desenvolupa un projecte és una bona manera de dominar-lo, però també es cometen moltes errades que s'han d'anar corregint al llarg del projecte.

Seguidament, la creació del personatge principal, concretament del component *PlayerController*, va suposar una gran dificultat per la complexitat del codi a desenvolupar. Inicialment, el moviment no era gaire realista i es va haver de fer que el personatge rellisqués en plans molt inclinats. A més, no s'havia de permetre que es pogués fer un salt quan el *Collider* fos a terra, perquè es podien pujar plans gairebé verticals si es saltava tota l'estona. Després de moltes consultes al fòrum de *Unity*, es va trobar una solució al problema. A més, es va trobar un *asset* a la *Asset Store* que oferia gratuïtament tota la funcionalitat de moviment d'un personatge en tercera persona i que es podia implementar perfectament en el personatge. Finalment, es va decidir que no calia afegir-lo perquè era molt complex i en el moment que hagués calgut fer modificacions, hauria sigut molt difícil per a nosaltres.

Tot i que s'han utilitzat molts *assets* gratuïts desenvolupats per tercers, ha sigut un gran repte trobar-ne de vàlids per utilitzar en el joc, ja que no n'hi ha gaires per triar i no tots els dissenys artístics encaixen els uns amb els altres. Al cap i a la fi, el disseny gràfic està fora de l'abast del projecte i per tant, s'ha hagut de confiar en aquells que hi havia a la *Asset Store* i similars.

Per acabar, cal mencionar el desenvolupament dels *tòtems* com un gran repte durant el projecte. La implementació ha passat per diferents iteracions, el funcionament de les quals era molt millorable. Quan es va aprendre un millor mètode durant la implementació d'una altra part del joc, es va retornar als *tòtems* per fer-los des de zero i aconseguir així un més bon resultat.

11. Futures millores

Tot i que el joc ja està finalitzat, hi ha molts aspectes que es poden canviar per millorar l'experiència de l'usuari, de manera que aprengui més i tingui més motivació per jugar.

Primerament, es podria treballar per fer el joc disponible a altres plataformes, com *Linux*, *Android* i *iOS* i així fer-lo més accessible, ja que no tothom té un ordinador amb la capacitat de fer córrer un joc amb gràfics 3D.

A més, seria convenient fer algunes optimitzacions per renderitzar alguns elements de la *Scene* i en el càlcul de físiques d'alguns objectes. N'hi ha alguns que no cal dibuixar en pantalla, o potser no amb tant de detall, si estan prou lluny. En la mateixa línia, també hi ha objectes que són estàtics i no necessiten passar pels càlculs de físiques que fa *Unity* a cada *frame*. Això pot estalviar recursos i millorar el rendiment del joc.

Un altre aspecte que té molt marge de millora són els tests interactius. Es poden crear nous tipus de tests creatius que aprofitin més l'entorn i interaccionin amb ell. A més, seria interessant augmentar el temari d'aprenentatge perquè els tests puguin ser de major interès i utilitat.

El *feedback*, és a dir, la informació que rep l'usuari en resposta a les seves accions, pot no ser suficient en alguns casos i el jugador pot veure's perdut en algunes situacions. Per això és importat assegurar-se que sempre es dona informació en forma de so, text i efectes especials, per tal que s'entengui allò que està passant en el joc en tot moment.

Finalment, segons els jugadors que han tingut ocasió de provar el joc, els controls del personatge han estat difícils i poc intuïtius, sobretot si es té en compte que el públic objectiu no és gaire experimentat en els videojocs. Així doncs, caldria fer que el moviment fos més senzill o informar millor al jugador de com moure el personatge correctament.

12. Planificació i pressupost

El procés de desenvolupament del projecte es pot categoritzar en diverses etapes i el seu corresponent temps d'execució, en hores. Tot i això, com hi ha dos membres al grup, s'ha pogut treballar en moltes etapes simultàniament. A més, la naturalesa del projecte fa que s'hagin de revisar parts antigues en algunes ocasions per corregir errors o afegir noves funcionalitats. Per això, la planificació compta també amb la despesa de temps a treballar en contingut teòricament finalitzat. Vegem tot seguit les etapes del projecte:

1. **Ideació i proposta del projecte (10 hores):** Proposta de la idea general del joc i la intenció de fer el projecte al professor.
2. **Estudi de mercat i públic objectiu (20 hores):** Anàlisi de l'estat de l'art i del públic a qui està dirigit el joc.
3. **Disseny conceptual del videojoc (40 hores):** En aquest punt, es tria el tema d'aprenentatge, es defineixen les mecàniques de joc i les característiques dels nivells i de l'entorn. També es fa un disseny preliminar dels menús, dels tests i es planteja la història del joc.
4. **Planificació de la implementació (15 hores):** Es planifica tota la implementació del videojoc mitjançant objectius generals i tasques setmanals que s'han de fer per aconseguir-ho a temps. A més, es realitza una assignació de tasques que es revisa al començament de cada setmana segons el progrés en cada moment.
5. **Aprenentatge de Unity i C# (40 hores):** Descàrrega i configuració del *motor de videojocs Unity* i aprenentatge d'ús del programa. Mentrestant, també s'aprofita per implementar a poc a poc alguna part del joc.
6. **Implementació del projecte (550 hores):** Engloba la creació del contingut del joc i es pot subdividir en les següents categories:
 - a. **Cerca d'assets (30 hores):** Es busquen els recursos artístics necessaris i es llisten per a un futur ús.
 - b. **PlayerController i càmera (80 hores):** Moviment i control del personatge i de la càmera.
 - c. **Tests (90 hores):** Es creen els tests a *Unity* per fer-los servir després en els *tòtems*.
 - d. **Tòtems (70 hores):** Implementació dels *tòtems* i tota la seva funcionalitat.
 - e. **NPC (45 hores):** Es descarreguen diferents models de personatges i es crea tota la funcionalitat de *NPC* per conversar amb el jugador i donar missions.
 - f. **UI (80 hores):** Disseny del menú del joc; engloba tot el contingut del menú de pausa (*Archive, Quests, Inventory, Map, Settings*) i el menú d'inici.
 - g. **Nivells (90 hores):** Creació de l'entorn de tots els sis nivells del joc: Cova, Bosc, Poble, Cementiri, Port i Castell.
 - h. **Emmagatzematge de dades (30 hores):** Es dissenya l'estructura dels fitxers

XML. També es desenvolupen els programes que els llegeixen i que escriuen sobre ells.

- i. **Música i Efectes de So (35 hores):** S'afegeix tot el contingut de música i efectes de so al joc i es crea el sistema de passes mitjançant un asset.
- 7. **Validació i correcció d'errors (45 hores):** Es crea el qüestionari de validació i es distribueix el joc a coneguts i amics. A més a més, s'aprofita per corregir els errors que notifiquen.
- 8. **Redacció de la memòria (75 hores):** Documentació de tot el procés de desenvolupament d'aquest projecte.

Així doncs, el total d'hores invertides en el projecte, entre els dos integrants, és d'aproximadament 795 hores en un període de quatre mesos. El diagrama de *Gantt* de l'annex [Annex 1] mostra amb més detall i de forma visual tot aquest procés de desenvolupament que s'ha exposat.

Per fer el càlcul dels costos del projecte, s'han considerat, per una banda, els costos de desenvolupament a partir de les hores invertides pels integrants del grup, com els costos addicionals que hi ha hagut en el procés i la depreciació d'algunes eines que s'han utilitzat.

Respecte a això, s'afegeix un cost de 5 € mensuals durant 4 mesos per la subscripció a *Unity Collab*, per fer servir el repositori del projecte i compartir-lo entre els integrants del grup a través d'internet. Això suposa un cost total addicional de 20 €. El pressupost del projecte es pot veure a la Taula 12.1.

Descripció	Temps [h]	Cost unitari [€/h]	Cost [€]
Gestor de projectes	205	20 [50]	4100
Dissenyador gràfic	170	10 [51]	1700
Programador	420	15 [52]	6300
Llicència de <i>Unity</i>	-	-	20
Depreciació i material	-	-	200
Total	795	-	12320

Taula 12.1 Pressupost del projecte

Si se suma tot, el desenvolupament d'aquest projecte suposa un cost total de 12320 €.

13. Impacte amb l'entorn

Aquest apartat avalua l'impacte sobre l'entorn que té l'ús del videojoc desenvolupat en aquest projecte.

Per una banda, es considera l'impacte social que té el programa. Es tracta d'un joc educatiu que intenta ajudar a la gent a entendre conceptes d'electrònica. Per això, l'impacte es considera positiu.

Per altra banda, s'ha de tenir en compte l'impacte amb el medi ambient a causa del consum energètic que suposa l'ús del programa per part de l'ordinador de l'usuari. Així i tot, s'ha considerat que aquesta despesa energètica és molt baixa i per tant, el seu impacte negligible. A més a més, no s'ha comprat cap material per a la realització del projecte i per a la comunicació i disseny s'han emprat sobretot eines digitals.

Conclusions

En retrospectiva, es pot dir que s'han realitzat de manera satisfactòria tots els objectius plantejats a l'inici d'aquest treball: dissenyar, implementar i validar un joc seriós en tres dimensions per a ordinador.

Prèviament a l'inici del projecte, s'ha dut a terme una anàlisi de mercat i d'usuaris. Aquesta etapa del projecte ha permès analitzar l'estat de l'art dels jocs seriosos i el seu públic potencial. Per consegüent, s'ha orientat tot el desenvolupament a una necessitat de mercat i s'han pogut cobrir els dèficits dels seus antecedents. Aquesta anàlisi de mercat ens ha ajudat a ser conscients de la importància de tenir en compte els usuaris potencials durant el desenvolupament d'un producte, no només en la informàtica, sinó en qualsevol àmbit de l'enginyeria.

En el transcurs del projecte, hem assumit diferents rols: dissenyador conceptual, dissenyador gràfic i programador. D'aquesta manera, ha estat possible aplicar tots els coneixements informàtics adquirits en la carrera i aprendre'n de nous, com el llenguatge *C#*. Així mateix, la realització d'aquest projecte també ha fet possible l'aprenentatge d'eines molt útils, com per exemple, *photoshop*, *inkscape* i, especialment, *Unity*.

La validació en finalitzar el treball ha estat una experiència molt positiva, ja que les opinions dels enquestats han tingut molta importància per a la detecció dels punts forts i febles de l'aplicació, així com per a la detecció d'errors.

En definitiva, la resolució de totes aquestes etapes ha menat cap a un joc complet i útil. Les proves de validació demostren que els estudiants estan motivats per aprendre i es diverteixen fent-ho. La informació docent necessària per a assolir cada objectiu és fàcilment accessible i la matèria incrementa la seva complexitat a un ritme adequat. Aquest resultat ha estat solament possible gràcies al treball en equip i ens ha aportat una gran experiència, tant professional com personal.

Agraïments

Volem agrair al nostre tutor Marc Vigo l'orientació i l'ajuda rebuda en el transcurs del treball, així com l'oportunitat de dur a terme aquest projecte.

També volem fer arribar el nostre agraïment a la professora Dani Tost pels seus consells i per la paciència que ha tingut amb nosaltres. Els seus suggeriments han estat cabdals en el desenvolupament del treball.

Així mateix, manifestem el nostre reconeixement a tots els artistes que lliuren les seves obres a la comunitat de manera gratuïta. L'estat actual del projecte s'ha assolit gràcies a les seves aportacions.

Finalment, apreciem el suport incondicional de la nostra família i amics, en especial de la Silvia Padrós, de la Laura Martos, de la Maria Monjarás i d'en Pol Jané. Sens dubte el projecte no hauria estat possible sense ells.

Bibliografia

- [1] HUOTARI, K. and HAMARI, J., 2012. Defining gamification. Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12, DOI 10.1145/2393132.2393137.
- [2] DETERDING, S., DIXON, D., KHALED, R. and NACKE, L., 2011. From game design elements to gamefulness. Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11, DOI 10.1145/2181037.2181040. BOSSER, J. *Vademécum de mecánica de fluidos y maquinas hidráulicas*, Barcelona: ETSEIB - CPDA . 1985.
- [3] ARA, 2016. Cap a la ludificació: viure jugant. Ara.cat [en línea]. Disponible en: https://www.ara.cat/premium/Cap-ludificacio-viure-jugant_0_1040296049.html. [28 de maig de 2018]
- [4] The (short) prehistory of "gamification"... Funding Startups (& other impossibilities) [en línea], 2012. Disponible en: <https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>. [28 de maig de 2018]
- [5] SINAN, M., 2010. The ultimate healthcare reform could be fun and games. VentureBeat [en línea]. Disponible en: <https://venturebeat.com/2010/04/12/healthcare-reform-social-games-gamification/>. [28 de maig de 2018]
- [6] HAMARI, J., KOIVISTO, J. and SARSA, H., 2014. Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification. 2014 47th Hawaii International Conference on System Sciences, DOI 10.1109/hicss.2014.377.
- [7] Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service. Egyptian Journal of Medical Human Genetics [en línea], 2013. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1567422313000112>. [28 de maig de 2018]
- [8] ZICHERMANN, G., 2011. Gamification by Design. S.I.: O'Reilly Media.
- [9] HAMARI, J. and KOIVISTO, J., 2015. "Working out for likes": An empirical study on social influence in exercise gamification. Computers in Human Behavior, vol. 50, pp. 333–347. DOI 10.1016/j.chb.2015.04.018.
- [10] HAMARI, J., SHERNOFF, D.J., ROWE, E., COLLIER, B., ASBELL-CLARKE, J. and

- EDWARDS, T., 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior*, vol. 54, pp. 170–179. DOI 10.1016/j.chb.2015.07.045.
- [11] Asociación Española de Videojuegos. Disponible en: <http://www.aevi.org.es/>. [10 de mayo de 2018]
- [12] Github – solubol/HalimdorCurse: 2D game created for university. Developed in python [en línea], Disponible en: <https://github.com/solubol/HalimdorCurse/> [6 de marzo de 2018]
- [13] Google Drive – Lord of Math. Desenvolupament d'una aplicació gamificada d'ajuda a l'estudi. [en línea], Disponible en: https://drive.google.com/open?id=1siyB7bqyGFaA6sF3EPAnwKLvILg_TdXH [30 de mayo de 2017]
- [14] Unity Asset Store - The Best Assets for Game Making. Asset Store [en línea], Disponible en: <https://assetstore.unity.com/> [9 de junio de 2018]
- [15] Mixamo. Mixamo [en línea]. Disponible en: <http://mixamo.com/> [9 de junio de 2018].
- [16] OpenGameArt - OpenGameArt.org. Disponible en: <http://opengameart.org/>. [9 de junio de 2018]
- [17] Your complete resource for training and development. Training Industry [en línea]. Disponible en: <https://directory.trainingindustry.com/training-supplier/gamelearn/>. [12 de junio de 2018]
- [18] Game-based learning formación corporativa. Gamelearn: Game-based learning courses for soft skills training [en línea]. Disponible en: <https://www.game-learn.com/game-based-learning-formacion-corporativa/>. [12 de junio de 2018]
- [19] Game-based learning formación corporativa. Gamelearn: Game-based learning courses for soft skills training [en línea]. Disponible en: <https://www.game-learn.com/game-based-learning-formacion-corporativa/> [12 de junio de 2018].
- [20] Precios e-learning competitivos y formación eficaz gracias al game-based training. Gamelearn: Game-based learning courses for soft skills training [en línea], 2017. [en línea]. Disponible en: <https://www.game-learn.com/precios-e-learning-gracias-al-game-based-training/>. [12 de junio de 2018]
- [21] Monster Guard. American Red Cross [en línea]. Disponible en: <http://www.redcross.org/monsterguard>. [12 de junio de 2018]

- [22] GARRIGA, P. and ARIADNA, 2017. Desenvolupament d'un joc seriós per a la formació de pacients amb diabetis tipus I. Pàgina inicial de UPCommons [en línia]. Disponible en: <https://upcommons.upc.edu/handle/2117/99233>. [12 de junio de 2018]
- [23] ITZIAR, 2017. Disseny i desenvolupament d'una aplicació per al tractament logopèdic. Pàgina inicial de UPCommons [en línia]. Disponible en: <https://upcommons.upc.edu/handle/2117/99037> [12 de junio de 2018]
- [24] Minecraft.Minecraft Education Edition [en línia]. Disponible en: <https://education.minecraft.net/>. [8 de junio de 2018]
- [25] ARTe – Triseum. Triseum [en línia]. Disponible en: <https://triseum.com/artemecenas/>. [12 de junio de 2018]
- [26] LLC, T., Triseum - Store. Triseum LLC [en línia]. Disponible en: <https://store.triseum.com/products/e5hts0>. [12 de junio de 2018]
- [27] VARIANT: LIMITS 2017A5 - POST GAME PLAY SURVEY REPORT [en línia]. Disponible en: https://s3-us-west-2.amazonaws.com/triseum-variant/collateral/Variant_Post-Gameplay-Results_SPR17.pdf. [12 de junio de 2018]
- [28] Gamification Market by Solution (Consumer driven and Enterprise driven), Applications (Sales and Marketing), Deployment Type (On-Premises and Cloud), User Type (Large Enterprise, SMBs), Industry and Region - Global Forecast to 2020. Market Research Firm [en línia]. Disponible en: <https://www.marketsandmarkets.com/Market-Reports/gamification-market-991.html>. [12 de junio de 2018]
- [29] Gamification Survey Results. TalentLMS Blog [en línia], 2017. [en línia]. Disponible en: <https://www.talentlms.com/blog/gamification-survey-results>. [12 de junio de 2018]
- [30] Unity. Unity [en línia],. Disponible en: <http://unity3d.com/>. [4 de junio de 2018]
- [31] Make Something Unreal. Unreal Engine [en línia], . Disponible en: <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>. [4 de junio]
- [32] Free 3D Game Engine. Panda3D [en línia],. Disponible en: <http://www.panda3d.org/>. [29 de mayo]
- [33] Familia de productos Adobe Photoshop. Adobe Captivate - Welcome to the world of smart eLearning authoring [en línia],. Disponible en: <https://www.adobe.com/es/products/photoshopfamily.html> [12 de junio]

- [34] Inkscape <https://inkscape.org/es/> [12 de junio]
- [35] Trello. Trello [en línea], Disponible en: <http://trello.com/>. [6 de junio]
- [36] Google [en línea],. Disponible en: <http://drive.google.com/>. [16 de junio]
- [37] Unity Services - Collaborate. Unity [en línea]. Disponible en: <https://unity3d.com/es/unity/features/collaborate>. [6 de junio]
- [38] Leonard's Blog Adventures in programming! *Leonard's Blog* [en línea. Disponible en: <http://www.lbd4.com/blog/>. [8 de junio]
- [39] PowerUp particles. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/vfx/particles/powerup-particles-16458> [10 de junio]
- [40] Rock and Boulders 2. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/props/exterior/rock-and-boulders-2-6947>. [11 de junio de 2018]
- [41] Cartoon Temple Building Kit Lite. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/dungeons/cartoon-temple-building-kit-110397>. [11 de junio]
- [42] Rocky Hills Environment - Light Pack. Asset Store [en línea], [sin fecha]. [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/landscapes/rocky-hills-environment-light-pack-89939> [11 de junio]
- [43] Nature Starter Kit 2. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977>. [11 de junio]
- [44] Mega Fantasy Props Pack. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/fantasy/mega-fantasy-props-pack-87811>. [11 de junio]
- [45] Make Your Fantasy Game - Fantasy Environment Assets. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/fantasy/make-your-fantasy-game-fantasy-environment-assets-7101>. [15 de junio]
- [46] Old Sea Port. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/3d/environments/old-sea-port-36897>. [11 de junio]

junio]

- [47] Free Footsteps System. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/tools/audio/free-footsteps-system-47967>. [14 de junio]
- [48] UET Particles Kit. Asset Store [en línea]. Disponible en: <https://assetstore.unity.com/packages/vfx/particles/uet-particles-kit-19690>. [11 de junio]
- [49] Formularios de Google: crea y analiza encuestas de forma gratuita. Google Search [en línea]. Disponible en: <https://www.google.es/intl/es/forms/about/> [15 de junio]
- [50] Sueldos en Gestor/a de proyectos en España [en línea]. Disponible en: <https://www.indeed.es/salaries/Gestor/a-de-proyectos-Salaries?period=monthly> [20 de junio]
- [51] Sueldos en Diseñador/a gráfico/a en España [en línea]. Disponible en: <https://www.indeed.es/salaries/Dise%C3%B1ador/a-gr%C3%A1fico/a-Salaries?period=hourly> [20 de junio]
- [52] Sueldos en Desarrollador/a de software en España [en línea]. Disponible en: <https://www.indeed.es/salaries/Desarrollador/a-de-software-Salaries?period=hourly> [20 de junio]

Annex

Annex 1. Diagrama de Gantt de la planificació temporal i desenvolupament del projecte.

